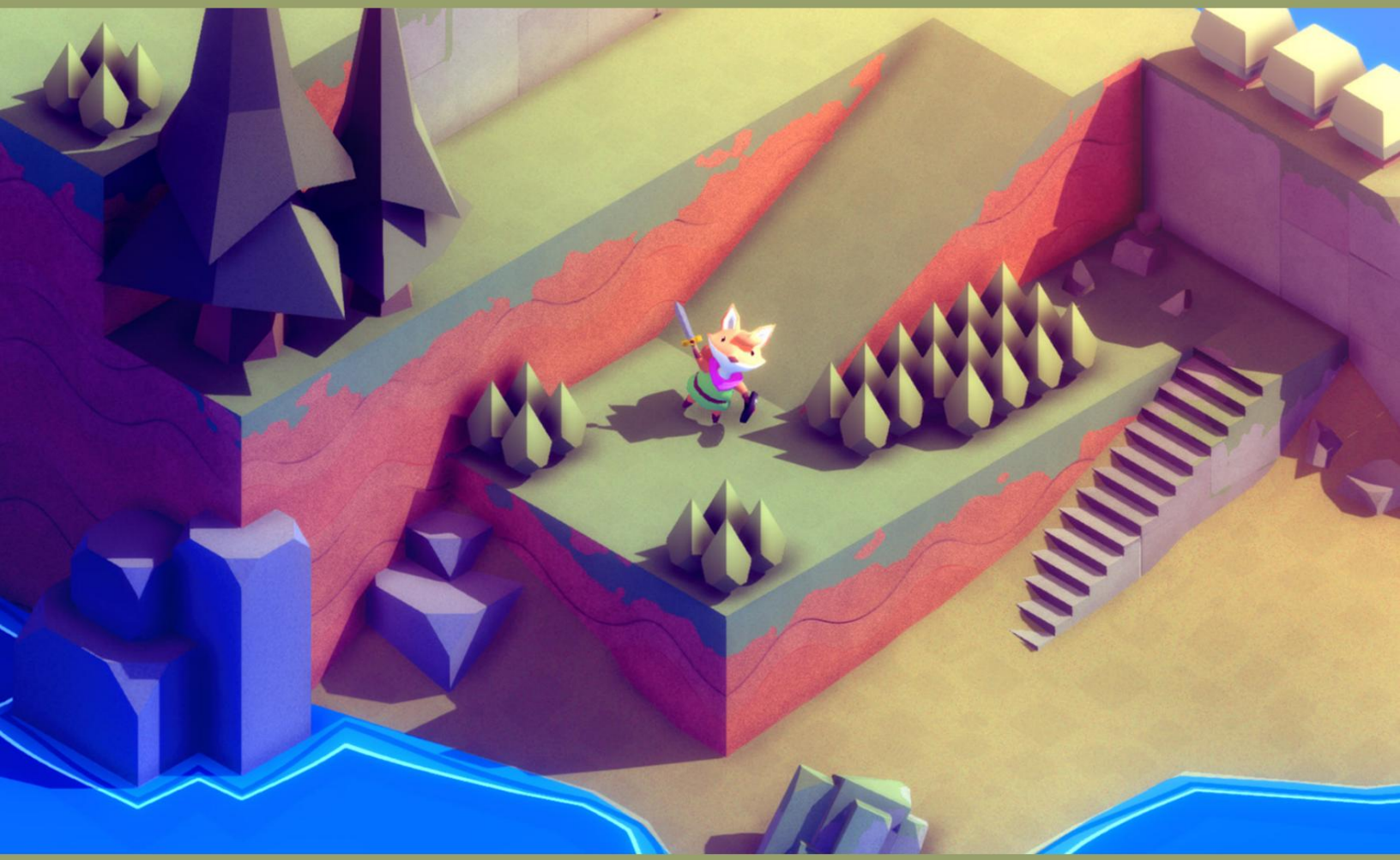
Machine Translated by Google

Unity®

LIBRO ELECTRÓNICO



INTRODUCCIÓN A DISEÑO DE NIVELES DE JUEGO

EDICIÓN LTS DE UNITY 2022

Contenido

Introducción. ...

Colaboradores. 4

Parte I: Introducción al diseño de niveles en el desarrollo de juegos. 5

Términos comunes utilizados en el diseño de niveles. 5

Preproducción... 8

Diseño coherente... 8

Guía de investigación y referencias 9

Conozca a su audiencia. 10

Las 3C: cámara, personaje y control... 12

Cámara. 12

Personaje. 12

Control. ...

Métricas 14

Diseño de papel. 16

Identificar y comprender las diferentes mecánicas del juego. . 17

Pon a prueba la mecánica del juego en un gimnasio/zoológico. 18

Ritmo y ritmo de juego. 19

Un mundo vivo: Narración ambiental 20

Producción. 21

Caja blanca: bloquear tu nivel. 21

Dar nombres descriptivos a los activos en forma de bloque. 22

Agregue materiales a los activos en bloques para

visualiza tu intención... 23

Evite crear recursos artísticos detallados 23

Pídele ayuda a un artista conceptual. 23

Los activos ya preparados también pueden ayudar. 24

Rutas del jugador... 25

Ruta crítica . . . 25

Camino dorado . . . 26

Caminos secundarios y terciarios: misiones secundarias, secretos y atajos. 26

La regla de tres para enseñar mecánica a los jugadores... 27

Subvirtiendo las expectativas... 28

Línea de visión y puntos ciegos. 29

Llamar la atención del jugador. 29

Iluminación y espacio. . . . 30

Bloqueadores físicos. 30

Señalización. . . . 31

Sonido. . . 32

Puntos de aparición 32

Puntos de guardado y puntos de control 32

Puntos guardados... 33

Puestos de control. . . 33

Evite el bloqueo suave... 33

Ubicación consciente de los puntos de guardado y puntos de control. . . . 34

Diseño a nivel procedimental. . . . 34

Reglas de generación procedimental . . 35

Pruebas automatizadas... 36

Prueba, prueba y vuelve a probar... 36

Parte II: Introducción a Unity para diseñadores de niveles. 37

Instalar Unity... 37

El editor de Unity. 39

Administrador de paquetes 41

Objetos del juego . . . 41

Manipulación de GameObjects en la vista de escena 42

Creando GameObjects. 44

GameObjects estáticos y dinámicos. 44
GameObjects activos/inactivos. 45
Etiquetas. 45
Prefabricados: GameObjects reutilizables. ...
3D o 2D. 48
Activos 3D 48
Formatos de archivos. 49
Activos 2D 49
Formatos de archivos. 50
Codificación. 51
Descripción rápida de C#. 51
Creando un nuevo script... 52
Un ejemplo de guión. 53
Recursos de aprendizaje. 54
Scripting visual de Unity. 54
Creando un nuevo gráfico de script. 55
Variables en Visual Scripting 56
Máquinas de estados con gráfico de estados. 58
Más recursos 59
Física 59
Creando colisiones... 60
Componente colisionador. 60
Componente de cuerpo rígido. 60
Desencadenadores de colisiones... 63
Capas de física... 63
Diseño de niveles con física. 64
Frecuencia de actualización de física. 65
Animación 66
El sistema de animación. 66
Ventana de animación... 67

Máquina de estados de animación... 68

Cronología... 69

Más recursos... 69

La tienda de activos de Unity... 70

Parte III: Herramientas de diseño de niveles en Unity. 71

Activos iniciales... 71

ProBuilder... 74

Grupos de suavizado... 88

Editor UV. 90

Ejercicio rápido de texturización. 91

Consejo: Codificación de colores para un diseño de niveles rápido. 93

Consejo: habilite las superposiciones de dimensiones para ProBuilder. 93

Compartir niveles de ProBuilder con artistas ambientales. 94

Pincel poligonal 95

Consejo rápido: visualizadores... 97

Texto de objeto 3D. 97

Icono personalizado para GameObjects 97

Estrías. 98

Terreno. 100

Mapa de mosaicos 2D... 102

Extras del mapa de mosaicos 2D... 104

Búsqueda de rutas con navegación IA... 105

La ventana de Navegación: Agentes y áreas. 106

Tres pasos para crear navegables

caminos con navegación IA... 107

Referencia rápida de herramientas de diseño de niveles. 109

Recursos adicionales de diseño de niveles. ...

Capacitación profesional para creadores de Unity. 111

Machine Translated by Google

INTRODUCCIÓN

Creadores de mundos virtuales y magos de la percepción espacial: Los diseñadores de niveles son
Los maestros constructores en el desarrollo de juegos, encargados del diseño coherente y
Composición de espacios jugables plenamente realizados.

Esta guía de tres partes está destinada tanto a principiantes como a experimentados.
diseñadores. Fue escrito por diseñadores de niveles y juegos, tanto dentro de Unity como
de equipos profesionales de desarrollo de juegos.

Unity es la plataforma de desarrollo de juegos más utilizada, por lo que obtener Unity
Las habilidades pueden abrir nuevas oportunidades para usted, incluso si trabaja en equipo.
Otro motor.

¿Qué hay en esta guía?

La Parte I ofrece una introducción a dónde y cómo se desarrolla el trabajo de los diseñadores de niveles.
Se integra en el ciclo de producción del desarrollo de videojuegos. Los temas tratados incluyen
Investigar y desarrollar una idea, crear prototipos de diseños con cajas blancas o grises y trabajar con
los elementos fundamentales del diseño de niveles, como
Caminos de los jugadores, iluminación, pistas y recolección de puntos.

La Parte II es una introducción al Editor de Unity, los componentes básicos para agregar
elementos e interactividad en una escena en Unity, y trabajar con activos,
Scripting, física y animación.

La Parte III proporciona instrucciones detalladas sobre cómo utilizar ProBuilder y
Conjuntos de herramientas Polybrush, una introducción al sistema Unity Terrain y
Recomendaciones de herramientas disponibles en Unity Asset Store.

© 2023 Unity Technologies

Machine Translated by Google

Monument Valley 2 de Ustwo Games es un juego de rompecabezas que hace un uso inventivo del espacio 3D.

Este libro electrónico complementa el manual del diseñador de juegos de Unity. que proporciona una Una visión más amplia de las herramientas de Unity para diseñadores de juegos y niveles, así como expertos Consejos para diseñar el juego.

Colaboradores

Cristo Nobbs Es un diseñador técnico de juegos senior que se especializa en sistemas. Diseño de juegos y Unity (C#). Christo ha estado usando Unity desde Unity 4 y Trabajó anteriormente como diseñador técnico de juegos en PUBG/PlayerUnknown Producciones.

Stefan Horvath Tiene más de 10 años de experiencia en el desarrollo de juegos. Industria con experiencia en diseño de niveles y juegos, así como en aseguramiento de la calidad. Ha trabajado en títulos como Star Citizen de Cloud Imperium Games, Dead by Daylight de Behaviour Interactive y 505 Games, y Tribes of Midgard de Norsfell Games Inc.

Eduardo Oriz es un gerente senior de marketing de contenido en Unity que tiene muchos Años de experiencia trabajando con equipos de desarrollo de Unity como las herramientas 2D Grupo para brindar contenido instructivo avanzado a desarrolladores y estudios de juegos.

© 2023 Unity Technologies

Machine Translated by Google

PARTE I: INTRODUCCIÓN DISEÑO DE NIVEL DESARROLLO DE JUEGOS

Términos comunes utilizados en el diseño de niveles

Términos de diseño de niveles	Descripción
3C	Esto significa personaje, cámara y control. Las 3C se utilizan para describir rápidamente lo complejo, relación de interdependencia entre los jugadores entradas y las reacciones de sus personajes en pantalla, y cómo ese personaje es capturado en el marco de la cámara del juego (Fuente: Pluralsight).
Asequibilidad	La cualidad o propiedad de un objeto que define su posibles usos o deja claro cómo puede o debe ser utilizado. (Fuente: Merriam-Webster).
AI	En los juegos, la IA (inteligencia artificial) es un elemento del juego. entidad cuya funcionalidad depende de código de computadora en lugar de intervención humana. Los PNJ son entidades comunes de IA.
Bloqueo	Un proceso de creación de niveles muy simple. Diseños ambientales utilizando geometrías primitivas. formas y arte del juego (similar a una maqueta).
Codificación de colores	Marcar elementos en una maqueta de diseño con diferentes colores como medio de identificación o categorización.

© 2023 Unity Technologies

Machine Translated by Google

Ultimate Chicken Horse de Clever Endeavour Games es un juego de plataformas para fiestas en el que tú y tus amigos construyen el nivel e intentan llegar al final.

Ruta crítica	Este es el camino más largo que el jugador puede tomar para Completa el juego.
Primero jugable	El primer jugable es una versión de un juego que Proporciona elementos de juego importantes y funcionales (y activos). A menudo se basa en el prototipo creado. en preproducción (Fuente: Wikipedia).
Campo de visión	En los videojuegos en primera persona, el campo de visión/ La visión (FOV) es la extensión del juego observable. mundo que se ve en la pantalla en un momento dado momento (Fuente: Wikipedia).
Meta	Los objetivos son los desafíos y actividades que enfrenta un jugador. debe completarse para ganar y/o concluir un juego. La colocación de objetivos a lo largo de un juego proporciona la Jugador con sentido de logro y Los motiva a continuar.
Camino dorado	Por lo general, este es el camino que ofrece la mejor elementos de juego, historia, recompensas y/o secretos (Fuente: tv tropes).
Interactuable	Un objeto con el que el jugador puede interactuar.

© 2023 Unity Technologies

Machine Translated by Google

Términos de diseño de niveles	Descripción
Línea de visión	La línea de visión (a veces abreviada como LoS) es La visibilidad (quién puede ver qué) en el juego campo de batalla en los juegos de guerra y de rol (RPG).
Malla	Una colección de vértices, aristas y caras que actúan como base de un modelo en un videojuego.
Minimapa	Un minimapa es un pequeño mapa de referencia que normalmente colocado en la esquina de la pantalla y utilizado para ayudar a los jugadores a navegar por el nivel del juego (Fuente: Blog de Sharp Coder).
Modificación	El modding en los videojuegos es el proceso por el cual los jugadores/ aficionados alterando uno o más aspectos de un juego, como como elementos visuales, personajes o comportamientos.
Búsqueda de caminos	La búsqueda de rutas es una técnica que utiliza la IA para navegar a través de entornos, generalmente mediante el uso de un malla de navegación, o navmesh.
Prueba de juego	Una prueba de juego implica jugar con cada nueva versión. de un juego para encontrar errores y garantizar la jugabilidad fluir y explorar oportunidades de mejora.
Prototipo	En el desarrollo de juegos, un prototipo es un ejemplo reducido de una "porción" o escena de un juego creado Para probar ideas fundamentales del juego y mostrarlas como Prueba de concepto para inversores y otros partes interesadas.
Legibilidad	La cualidad de ser legible o descifrable.
De arriba hacia abajo	La perspectiva de arriba hacia abajo (o vista de pájaro o La vista de helicóptero en los videojuegos es la perspectiva del jugador mirando hacia abajo a un nivel desde arriba. La vista de arriba hacia abajo también se puede utilizar para describir una Visualización 2D de un nivel que se ve desde arriba.
Punto de vista	Una posición o punto de vista desde el cual se dice algo visto o considerado (Fuente: Merriam-Webster)
Estructura alámbrica	Un modelo de estructura alámbrica es una representación visual de un Objeto físico 3D utilizado en gráficos de computadora 3D (Fuente: Wikipedia).
Caja blanca (o caja gris)	Una caja blanca es un nivel creado con 3D simple formas que se utilizan para probar diseños y rutas.

Puedes encontrar más nomenclatura para juegos en el glosario. en la unidad documentación.

© 2023 Unity Technologies

Machine Translated by Google

Preproducción

Esta sección abarca algunos de los temas y tareas clave en la etapa de preproducción del desarrollo de videojuegos, como las filosofías de diseño, las métricas, los diseños en papel y la planificación de las mecánicas del juego. Si se abordan con cuidado, estos procesos pueden ayudarte a definir tu visión del juego y a avanzar hacia la producción de forma eficiente.

Diseño coherente

Un diseño de niveles coherente resulta de desarrollar una comprensión holística del mundo del juego que se está creando: su temática, estilo, época, etc. Investigar estos elementos, incluyendo la colaboración con colegas para desarrollar la temática y la historia del juego, así como sus métricas y mecánicas clave, contribuirá en gran medida a ofrecer a los jugadores un mundo inmersivo y rico en detalles.

En la escritura, la coherencia se crea asegurando que cada oración y párrafo se conecte con el siguiente en una secuencia lógica y fluida. De igual manera, el diseño coherente en el desarrollo de videojuegos busca ser sistemático y consistente en su presentación, con cada elemento del diseño del juego interconectado lógicamente, contribuyendo así a una experiencia holística.

Death's Door de Acid Nerve demuestra una excelente ejecución del diseño de niveles interconectados. Puedes ver la entrevista de Unity Creator Spotlight con el equipo aquí.

© 2023 Unity Technologies

Machine Translated by Google

La puerta de la muerte y Cuphead son ejemplos de juegos con un diseño sólido, bien ejecutado y coherente. Los desarrolladores afirman que El diseño de niveles de Death's Door se inspiró en la serie Dark Souls .

“Una de las cosas de Dark Souls que nos impresionó la primera vez que lo jugamos fue el espacio tridimensional que usan para los niveles”, explica Mark Foster (programador principal, diseñador de código, animador y escritor). “Todo está apilado, así que es genial encontrar un atajo”.

La interconexión de los niveles ayuda a sumergir al jugador en un mundo de juego que se siente como un espacio real, uno que también proporciona momentos de deleite, como cuando el jugador se da cuenta de que los niveles están dispuestos de manera orgánica y no lineal.

Y en Cuphead, todos los componentes trabajan juntos para crear una experiencia cohesiva, desde los aspectos visuales hasta la mecánica del juego y el sonido.

Guía de investigación y referencias

Reúne tu investigación y referencias en un documento o guía para comprender mejor el mundo del juego y los niveles que estás creando. Una guía de referencia organizada y actualizada periódicamente ayudará a todo el equipo de desarrollo a mantenerse al día en la ejecución de la estética del juego.

Cuphead de Studio MDHR, está inspirado en gran medida en los dibujos animados de la década de 1930.

© 2023 Unity Technologies

Machine Translated by Google

Pureref, Figma, y Pinterest Existen algunas aplicaciones para recopilar y compartir imágenes de referencia. Entre las aplicaciones y sitios web que pueden servirte de referencia para tus ideas de diseño de niveles se incluyen:

— Mapa VG: Un archivo de mapas de juegos clásicos, en su mayoría 2D

— Sin clip: Un archivo de mapas de niveles clásicos en 3D

— Sistema de información geográfica (SIG): Es un tipo de base de datos
Contiene datos geográficos, combinados con herramientas de software para gestionarlos, analizarlos y visualizarlos. Utilice datos SIG como base para su mapa de terreno y altura, tomando como referencia el mundo real y su distribución natural, y luego desarrolle a partir de ellos. Existen diversas herramientas disponibles para usar datos SIG en Unity, como Terraworld Automated Level Designer. desde la tienda de activos de Unity.

Terraworld Automated Level Designer de GISTech es un complemento de Unity que utiliza referencias del mundo real para generar terreno.

Además, busca inspiración y referencias en otros juegos. A veces, una idea puede ser más atractiva y comprensible si se utiliza una referencia funcional que la gente pueda jugar o ver.

Conozca a su audiencia

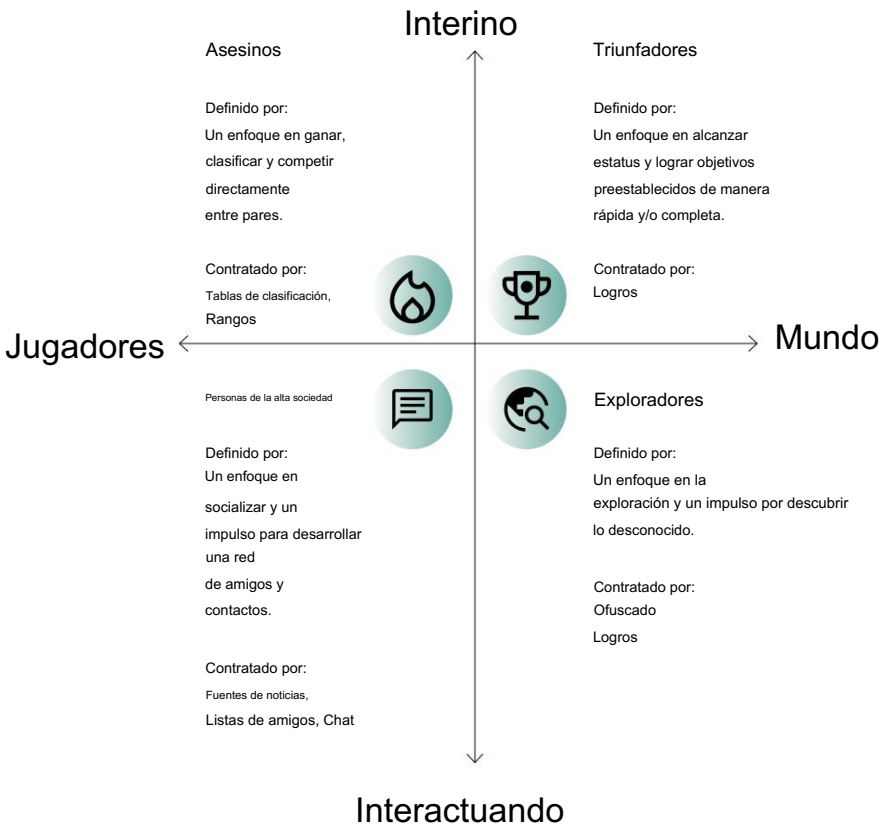
Conocer a tu público es importante para planificar tus niveles. ¿Te diriges a un público gamer exigente? ¿A un grupo de edad o estilo de juego específico?

Normalmente, el público objetivo lo definen el director creativo, el equipo de producción y el de marketing. Consulta con tus compañeros de estos equipos para comprender su perspectiva sobre el público objetivo del juego.

© 2023 Unity Technologies

Machine Translated by Google

Es útil conocer la taxonomía de Bartle, Clasificación de los jugadores de videojuegos según sus acciones preferidas. Se basa en la teoría de personajes, que identifica cuatro tipos: Triunfadores, Exploradores, Socializadores y Asesinos.



La taxonomía de Bartle es una clasificación de jugadores de videojuegos basada en un artículo de Richard Bartle de 1996. La fuente de la imagen está aquí.

Un diseñador de niveles puede construir el mundo con estos arquetipos de personajes en mente.

¿Tu juego es un multijugador competitivo por equipos? Entonces, tus arquetipos de personaje podrían ser Asesinos y Sociales, ya que estos juegos suelen centrarse en la competición entre pares y el trabajo en equipo.

Si tu juego es una aventura de mundo abierto para un solo jugador, tus arquetipos de personajes podrían ser Exploradores y Triunfadores, ya que el jugador puede explorar el mundo abierto mientras completa objetivos.

Un juego no tiene por qué centrarse únicamente en un arquetipo de personaje; existen maneras de incluir todos los arquetipos en la mayoría de los juegos. Por ejemplo, un juego para un jugador puede incluir el arquetipo de sociable, añadiendo una forma de interactuar con otros jugadores fuera del juego, intercambiando regalos o visitando la base de un jugador. Un juego de disparos multijugador, aunque se centre principalmente en el arquetipo de asesino, también puede ser atractivo para los triunfadores.

En última instancia, debes intentar mantenerte cerca de la visión del juego y satisfacer los arquetipos de personajes principales o más importantes y evitar diluir la jugabilidad para adaptarla a demasiados tipos de jugadores.

© 2023 Unity Technologies

Machine Translated by Google

Las 3C: cámara, personaje y control

Normalmente, un diseñador de niveles ajusta y perfecciona la cámara, los personajes y los controles de un juego mientras crea un gimnasio de pruebas para facilitar su trabajo. Este proceso puede ayudarle a comprender cómo se configuran las 3C en el juego, lo que en última instancia le ayuda a sentar las bases para las métricas clave.

Dado que tendrás una participación práctica en la enseñanza y presentación de estos movimientos al jugador, intenta ofrecer comentarios a los diseñadores del juego para mejorar la experiencia.

Cámara

La cámara ayuda a impulsar tus métricas y proporciona la perspectiva que usará el jugador para jugar el juego.

Algunos ejemplos de cómo un diseñador de niveles puede considerar la cámara durante la producción incluyen:

— Usar la altura de la cámara para evitar cortes cuando el jugador cruza puertas

— Usar el campo de visión (FOV) y la cámara para ilustrar la perspectiva del jugador y cómo se encuadrarán las escenas en el juego.

— Sopesar los pros y los contras de diferentes perspectivas: una historia en primera persona La perspectiva hace que el jugador mire en una dirección, lo que limita su capacidad de detectar enemigos que se acercan sigilosamente por detrás; en comparación, en un juego isométrico, el jugador normalmente ve todo a su alrededor.

Personaje

El elemento de personaje es importante para un diseñador de niveles porque se refiere a sus armas, habilidades y otros rasgos únicos. En definitiva, es la herramienta que define cómo el jugador navegará e interactuará con sus niveles.

La posición de la cámara y el campo de visión (FOV) dan una idea de la escala de los personajes y los entornos. En realidad virtual, esto es aún más importante, ya que te sumerges en el entorno. Aprende más sobre realidad virtual con el tutorial "RV Principiante: La Sala de Escape".

© 2023 Unity Technologies

Machine Translated by Google

Toma nota de cualquier cambio en las habilidades y otros rasgos del personaje durante el desarrollo, ya que pueden afectar la forma en que se juega el nivel.

Algunos ejemplos de cómo un diseñador de niveles puede tener en cuenta al personaje incluyen considerar:

— Las habilidades del personaje y cómo enseñarlas y mostrarlas al público.
jugador de forma gradual

— La fuerza del personaje en cualquier punto del juego y si los desafíos en las distintas etapas
son adecuados a su nivel de fuerza/poder.

— Opciones de personalización de personajes para el jugador y su forma de jugar, por ejemplo,
Sigilo, correr y disparar, piratería, etc.

— La capacidad de doble salto del jugador: ¿Qué tan alto y lejos puede saltar? ¿Está aprovechando
bien esta habilidad en sus diseños?

Control

El control determina cómo el jugador manipula a su personaje usando el periférico que prefiera. Ya sea navegando sigilosamente por los niveles, galopando entre árboles, corriendo en un coche o chocando como los personajes de Fall Guys, un buen control influye significativamente en la jugabilidad.

Juega a través de tus niveles con los diferentes periféricos objetivo para tu juego. Dependiendo de su audiencia, es posible que desee reducir la repetición de acciones en una secuencia corta o ampliar los pasillos para compensar la diferencia en los controles.

Slime Rancher 2 de Monomi Park es un trepidante juego de aventuras en primera persona. Como en cualquier FPS, los controles ágiles para girar, apuntar y moverse son clave para la experiencia de juego.

© 2023 Unity Technologies

Machine Translated by Google

Por ejemplo, la navegación con ratón y teclado permite movimientos precisos y un mayor rango de movimiento que con un joystick analógico, a menos que los valores estén fijados. El joystick analógico suele permitir moverse entre un valor de 0 a 100 a una velocidad fija, lo que puede resultar en velocidades de giro y tiempos de reacción más lentos en comparación con el ratón y el teclado.

Obtenga más información sobre la cámara, personaje, y controles de Unity Learn y documentación.

Métrica

Las métricas del juego determinan las relaciones en el espacio jugable. Es importante identificarlas y definirlas antes de comenzar la producción, ya que determinarán la composición del nivel y los espacios.

Por ejemplo, un personaje de movimiento rápido podría necesitar más espacio jugable que uno lento. Idealmente, el diseñador de niveles tendría la mayoría de las métricas a mano antes de crear su nivel y luego las probaría en un "zoológico", un espacio de juego para explorar los componentes del juego.

Otro ejemplo es el de un juego en el que un jugador puede agacharse y esconderse del enemigo tras objetos. Solo para ese sencillo mecanismo, necesitarías conocer las siguientes métricas:

— ¿Qué altura tiene el jugador cuando está agachado? (altura vertical del personaje modelo)

— ¿Qué altura tiene la cápsula/hitbox del jugador cuando está agachado (un espacio generalmente más grande o igual que el modelo del personaje que se usa para determinar los “golpes” al jugador)?

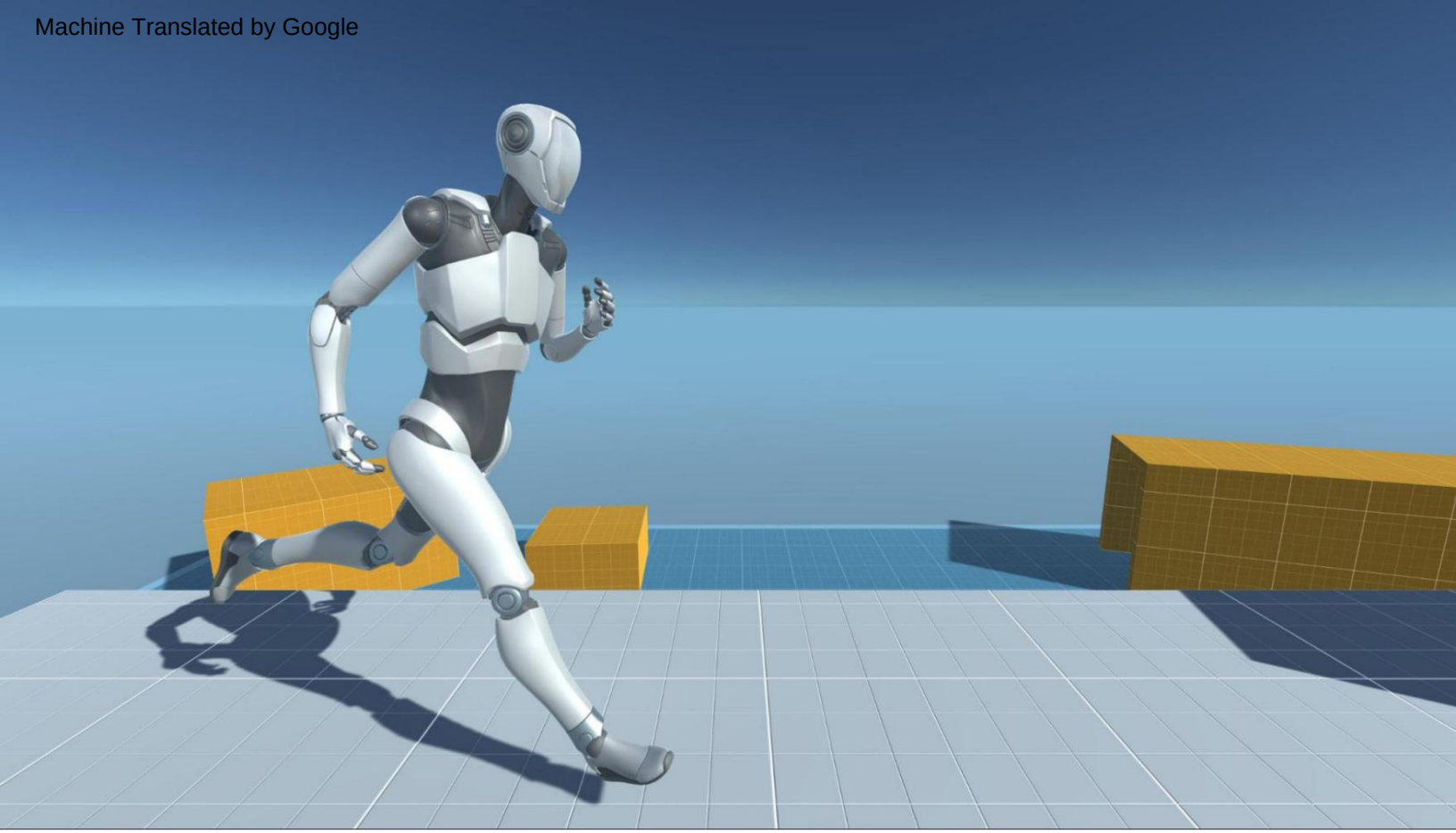
Usando estas dos métricas, puedes determinar la altura mínima de un obstáculo para que el jugador pueda esconderse del enemigo detrás de él y bloquear los ataques enemigos desde el lado opuesto del obstáculo.

Tenga siempre en cuenta los cambios en las métricas durante todo el ciclo de desarrollo del juego. En este ejemplo, si las métricas cambiaran después de completar los niveles, probablemente sería necesario revisar todos los obstáculos de cobertura, ya que ya no respetarían las reglas originales. Los cambios podrían dar resultados contrarios a los previstos, como que el jugador sea visto o golpeado cuando debería estar oculto.

Normalmente, el software 3D y los motores de juego como Unity se diseñan con un tamaño de unidad predeterminado que sirve de referencia para otros sistemas, como la física, las cámaras o el renderizado, para simular la realidad. Una unidad de Unity equivale a un metro, y es este tamaño de cuadrícula el que se ve en la vista de escena. Al crear un nuevo objeto 3D primitivo, como un cubo, se ocupa un metro cúbico de volumen. Es recomendable respetar esta escala al crear entornos y personajes.

© 2023 Unity Technologies

Machine Translated by Google



Activos iniciales Es una plantilla de Unity disponible en Unity Asset Store que incluye un controlador de personaje y una configuración de cámara. El nivel de prueba incluye una textura de cuadrícula que sigue la referencia de tamaño de unidad de Unity.

Al crear entornos 2D en Unity, Se recomienda mantener la escala relativa al mundo real, a pesar de ser más flexible. Por ejemplo, si tu juego 2D usa un sistema de mosaicos, puedes usar esa referencia, siempre que el resto del juego se diseñe en torno a ella.

Vea cómo modificar la cuadrícula del juego de Unity en la sección Primeros pasos en Unity a continuación.

Skul: El cazador de héroes es un juego rogue-lite de pixel art creado en Unity que utiliza el editor de mapas de mosaicos 2D para construir el mundo.

© 2023 Unity Technologies

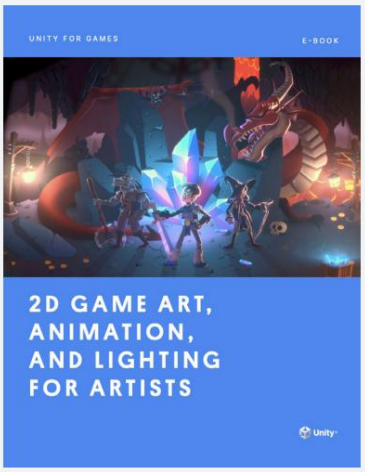
Machine Translated by Google

Arte de juegos 2D, animación e iluminación para artistas

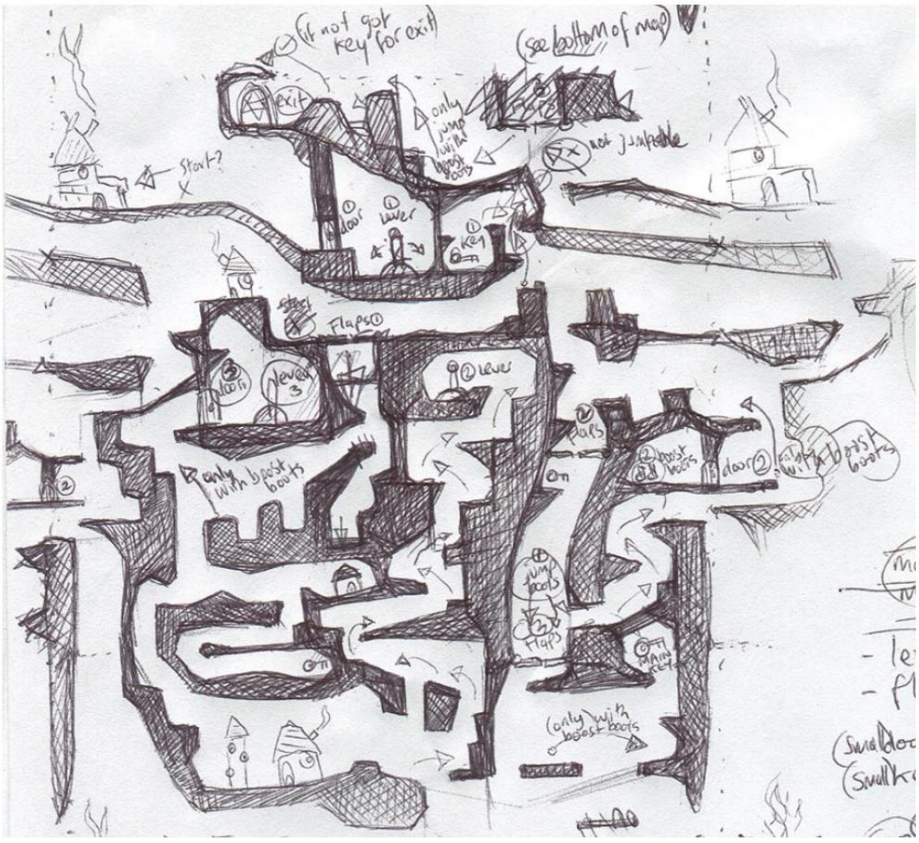
Los juegos 2D están dejando su huella. La evolución del hardware, los gráficos y el software de desarrollo de juegos hace posible crear juegos 2D con luces en tiempo real, texturas de alta resolución y un número de sprites casi ilimitado.

Obtenga la guía de desarrollo 2D más completa de Unity, creada para desarrolladores y artistas que desean crear un juego 2D comercial.

Descargar el libro electrónico



Diseño de papel



Un nivel dibujado a mano para un juego de plataformas 2D
Fuente: Peter Mcclory

Machine Translated by Google

Diseñar tus niveles en papel antes de ponerlos en Unity puede ser útil como
parte de la fase de investigación de creación de un nivel.

Un boceto o diseño aproximado de un nivel puede incluir los rompecabezas, los encuentros con los enemigos,
y otros elementos de juego que el jugador encontrará.

Diseñar en papel puede permitirle prever tanto problemas potenciales como
Beneficios con tus ideas de juego y la colocación de elementos. Es especialmente útil
Si no hay un controlador de personajes o un editor de niveles disponible en el motor del juego para
Bloqueo y prueba.

Por otro lado, no todos los diseñadores de niveles usan papel primero. Algunos prefieren trabajar
directamente en el motor de juego elegido, ya que es más fácil de visualizar y mover
alrededor del espacio.

Hay varias aplicaciones de dibujo que puedes utilizar, como por ejemplo Diagramas.
neto, Microsoft Visio, sí, o Gravit Designer.

Un nivel bloqueado con LEGO, por Hugues Barlet
Fuente: https://www.gamedeveloper.com/design/block-design-in-level-design

Identificar y comprender las diferentes mecánicas del juego.

La mecánica del diseño del juego y la jugabilidad que utilizan los jugadores son un factor importante.
parte de tu conjunto de herramientas. Cuanto más las aprendas y comprendas, más efectivas serán.
Lo estarán cuando los uses.

Por ejemplo, una mecánica de salto se puede dividir en varias partes dependiendo
sobre su implementación.

Lo más simple puede ser un salto estacionario: sin usar otros botones, un jugador
Puedes presionar el botón de salto para saltar en el aire. Una maniobra más complicada puede...
correr a toda velocidad, saltando a la altura máxima manteniendo presionado el botón de salto,
y realizar saltos de pared.

Una vez que comprenda las complejidades involucradas en la ejecución de la mecánica del salto,

© 2023 Unity Technologies

Machine Translated by Google

Puedes desglosar el resultado final y presentar a los jugadores cada mecánica gradualmente, comenzando con la acción más fácil de realizar y luego trabajando en interacciones más complejas a medida que el jugador progresa.

Utilicemos un salto de pared como ejemplo y dividámoslo en los siguientes segmentos:

— Presione el botón para saltar (simple salto hacia arriba, gana altura Y)

— Mantenga presionado el botón para saltar (salte más alto hacia arriba y gane altura Y + altura adicional).

— Correr y saltar, (ganar Y altura y X distancia)

— Corre y mantén el salto (gana altura Y + altura extra y X + distancia)

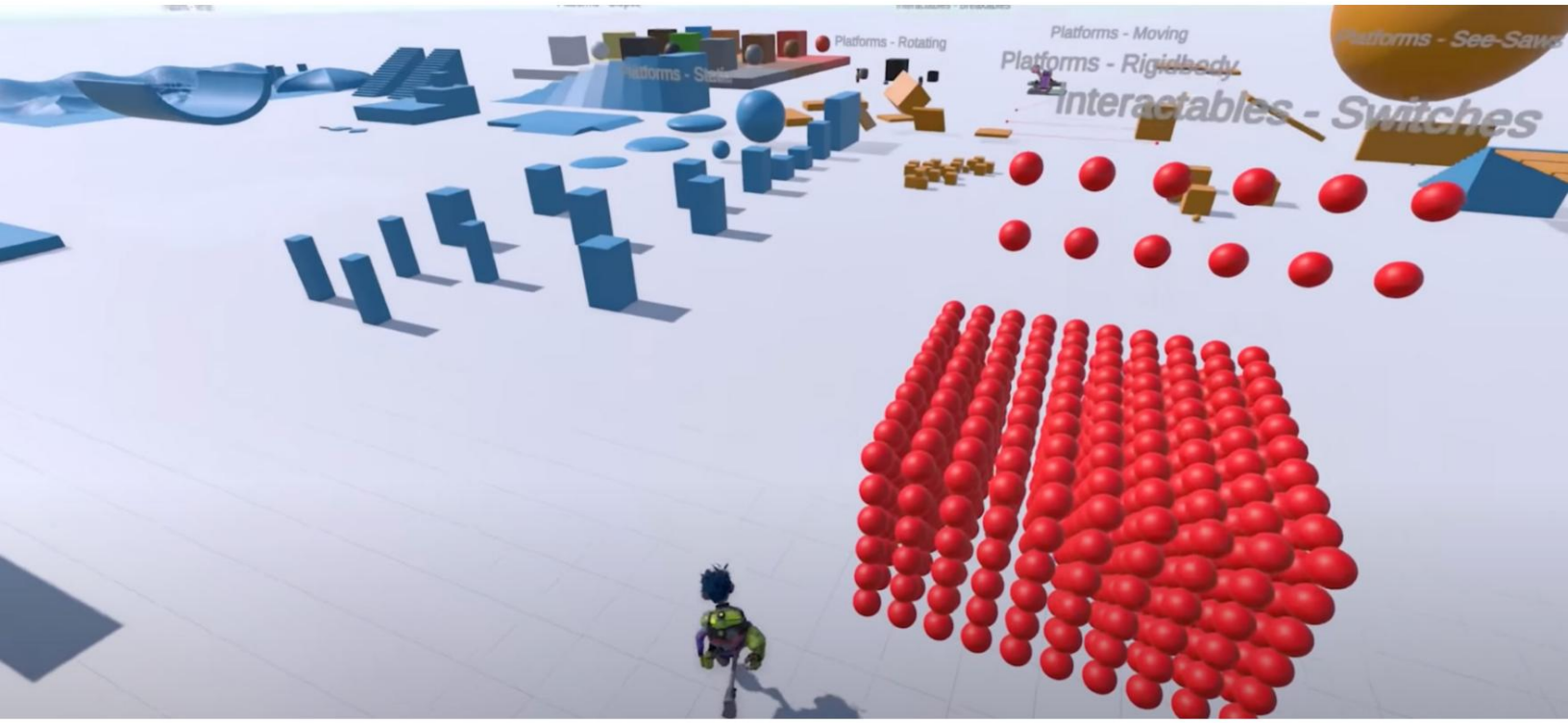
— Salta a una pared y deslízate hacia abajo.

— Saltar a una pared, deslízate hacia abajo y vuelve a saltar a otra pared, etc.

Cada uno de estos segmentos se puede presentar al jugador a medida que progresa, lo que hace que la curva de aprendizaje sea más manejable y, con suerte, ayuda a los jugadores a dominar cada habilidad.

Indivisible Es un juego que aprovecha al máximo la introducción gradual del jugador a las mecánicas de transición de niveles, comenzando con un simple salto y culminando con una combinación de varias mecánicas. Ofrece amplio margen para que el jugador aprenda y domine estas habilidades sin sentirse abrumado.

Prueba la mecánica del juego en un gimnasio/zoológico



Un nivel de gimnasio hecho con elementos intratables del prototipo y obstáculos.

© 2023 Unity Technologies

Machine Translated by Google

Un gimnasio, o zoológico, es un espacio de juego sencillo creado para probar y perfeccionar las mecánicas que se utilizarán en la versión final. Normalmente, el equipo de producción solo utiliza los gimnasios durante la producción.

¿Qué mecánicas puedes probar en un gimnasio?

— Crea puertas para probar la altura de la cámara y el espacio necesario a cada lado del jugador.

— Agregue giros, vueltas y caminos sinuosos para probar el movimiento y el control del jugador.

— Agregue pendientes para probar inclinaciones, incluidas múltiples inclinaciones para probar la granularidad

— Pruebe pendientes insalvables para equilibrar el terreno transitable e intransitable.

— Prueba disparadores, distancias de salto y mucho más

En última instancia, tu objetivo es identificar qué mecánicas y elementos de juego funcionan y cuáles deberías eliminar.

Si es necesario, puedes crear varios gimnasios para probar y aislar mecánicas, por ejemplo, para probar tipos de terreno, galerías de tiro, mecánicas de batalla, etc.

Comparte tus gimnasios con el equipo de desarrollo. El equipo de control de calidad, por ejemplo, puede usarlos para probar y aislar mecánicas, los diseñadores de juegos para ajustar, mejorar y probar sus diseños, y los artistas técnicos para probar recursos artísticos (niveles de detalle, etc.).

Unity proporciona un controlador de personaje que se puede usar como sustituto para probar las mecánicas. Si reemplazas el controlador de personaje de Unity por uno nuevo, deberías volver a probarlo en el gimnasio para asegurarte de que funciona con las métricas y funcionalidades del juego del controlador anterior y ajustarlo según sea necesario. Consulta la sección sobre controladores de personaje para obtener más información.

Ritmo y ritmo de juego

Un buen ritmo mantiene al jugador enganchado. Se apoya en los ritmos de juego, los puntos y actividades principales que impulsan al jugador a avanzar.

Un diseñador de niveles normalmente tiene control sobre el ritmo a lo largo del juego.
Crear una línea de tiempo visual puede ser útil para determinar qué tipo de ritmo quieres que los jugadores logren a medida que progresan y en qué puntos el jugador alcanzará otro ritmo de juego.

Dale libertad al jugador para explorar un nivel o parte del juego eliminando el límite de tiempo para esa parte en particular. Por el contrario, si necesitas que el jugador tenga sentido de urgencia, establece un límite de tiempo para que avance.

Podría valer la pena dar a tus jugadores un respiro en partidas intensas, permitiéndoles relajarse antes de volver a jugar con intensidad. Un ejemplo de esto es la fase de "accidente de coche" en Street Fighter II. Tras un intenso combate contra la IA u otros jugadores, el jugador puede destrozar un coche que no se defiende, lo que le permite practicar sus movimientos y recuperar la confianza.

© 2023 Unity Technologies

Machine Translated by Google

Sons of the Forest de Endnight Games (publicado por Newnight) es un popular juego de supervivencia que incluye la creación como una de las principales mecánicas del juego, una tarea que ofrece un agradable descanso entre peleas tensas.

Obtenga más información sobre el ritmo en este Vídeo de Brackeys.

Un mundo vivo: Narración ambiental

Harold Halibut de Slow Bros, es un juego narrativo hecho a mano sobre la amistad que hace un gran uso de Unity Cinemachine y Timeline.

Un hombre en un espacio con poca luz observa un grafiti en la pared. En rojo, dice "¿DÓNDE ESTÁ LA CASA?". A primera vista, parece escrito con sangre, pero una lata de pintura roja atrae su atención (y la del jugador). Esta escena invita al jugador a comprender lo que ve, mientras imagina su significado. Elementos ambientales sencillos como este grafiti y la pintura ayudan a sumergir a los jugadores en la historia del juego.

La narrativa ambiental consiste en contar una historia mediante el diseño de niveles y los recursos del entorno. Explica lo que ocurre en el juego con marcadores, pistas, etc., implícitos, en lugar de explícitos.

© 2023 Unity Technologies

Machine Translated by Google

Una construcción de mundos eficaz utiliza la narrativa ambiental para apoyar la narrativa del juego, creando un mundo auténtico y vivido que fomenta la inmersión del jugador. Los diseñadores de niveles pueden usar iluminación, recursos temáticos y accesorios cuidadosamente colocados para revelar la historia a los jugadores a medida que avanzan en el juego.

Aclare su narrativa en un juego respondiendo preguntas fundamentales como:

¿ Por qué sucede esto?: ¿Hay elementos que puedas incluir para dar pistas sobre por qué el jugador se encuentra en un punto determinado? Piensa en la progresión de la misión o la jugabilidad que lleva al jugador a un punto o hito específico y úsalo para definir el tono o la atmósfera de una escena o nivel.

— ¿ Qué pasó?: ¿La escena refleja el tono y los elementos que deben guiar la historia? ¿Para que el jugador comprenda lo que pasó?

— ¿ Dónde sucedió esto?: Piense en la ubicación, su historia de fondo y qué elementos deberían incluirse o destacarse para respaldar el lugar donde sucede esto.

— ¿ Cuándo sucedió esto?: Considere la línea de tiempo y el período del mundo del juego. Identificar cuándo sucedió ayuda al diseñador a ubicarse en la línea de tiempo y encontrar elementos a incluir para contar mejor la historia.

¿ Cómo sucedió esto?: ¿ Hubo un terremoto que causó pánico en toda la ciudad? Piensa en cómo puedes mostrárselo al jugador con elementos de fondo, como imágenes caídas, grietas en los edificios, etc.

Responder estas preguntas puede ayudar a completar los espacios en blanco al desarrollar tus niveles, lo que hará que sea más fácil tener un mundo cohesivo y completo construido sobre una historia de fondo lógica.

Producción

Tras la preproducción, deberías contar con las herramientas y la información necesarias para empezar a planificar e iterar tus niveles. Esta sección ofrece consejos y prácticas recomendadas para desarrollar niveles.

Caja blanca: bloquear tu nivel

El white-boxing (también llamado grey-boxing) consiste en crear y organizar formas 3D simples para identificar qué diseño se adapta mejor al diseño de nivel y al estilo que buscas.

Mantener la caja blanca simple le permite manipular fácilmente el nivel sin tener que ajustar el arte, la iluminación y otros detalles, lo que resulta en un proceso de iteración más rápido.

Algunos diseñadores de niveles prefieren empezar directamente con el diseño en caja blanca en lugar de crear primero diseños en papel, ya que les gusta trabajar en el espacio 3D y familiarizarse con los flujos de trabajo del motor de juego. Tú decides qué enfoque te funciona mejor.

La Parte III de esta guía proporciona información paso a paso sobre cómo utilizar ProBuilder de Unity. Para generar formas y terreno de forma eficiente. Ofrece herramientas de modelado listas para usar y elementos predefinidos comunes en el diseño de juegos. Como alternativa, puedes colocar objetos 3D primitivos directamente en la vista de Escena, un proceso que se explica en la Parte II.

© 2023 Unity Technologies

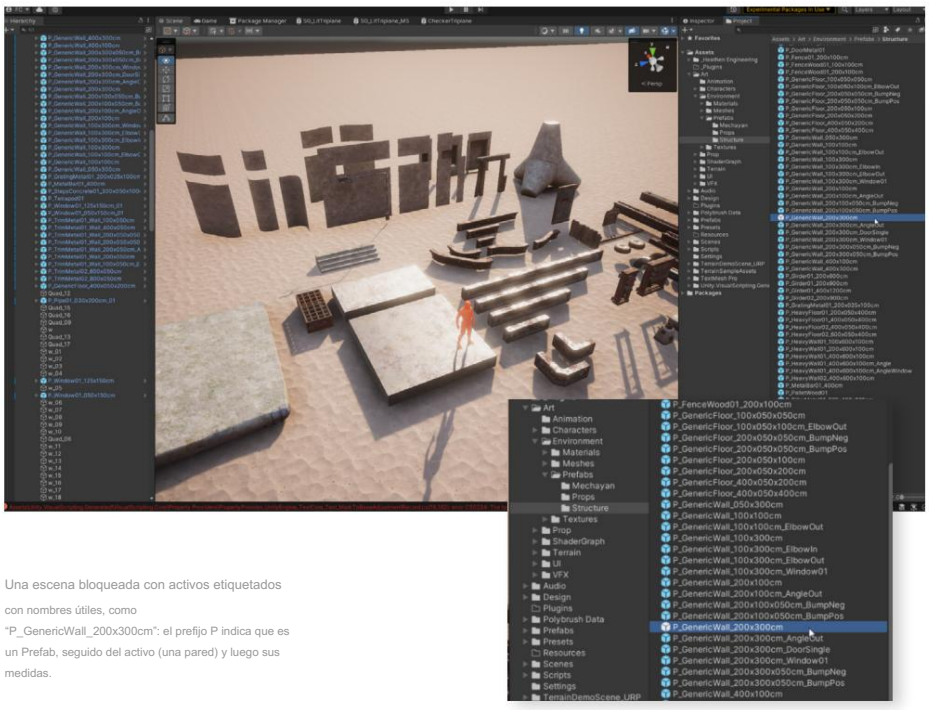
Machine Translated by Google

Shooter con múltiples armas, TDDM, IA y realidad virtual en D18 (reddit, publicado por niv-vada). Ejemplo de un nivel bloqueado con ProBuilder.
Hilo de Reddit

Dar nombres descriptivos a los activos en forma de bloque

Asigna un nombre descriptivo a cada elemento de bloque para identificar su uso. Esto ayuda a los diseñadores de entornos a comprender tu intención para cada objeto cuando necesiten reemplazarlo con los elementos del juego.

Por ejemplo, ¿es un bloque una pared? De ser así, ¿es necesario que tenga una altura mínima para bloquear la vista del jugador? Un nombre para este recurso podría ser "pared_interior_ w2_h4_l6". Esta etiqueta identifica el objeto, su ubicación y medidas, detalles importantes que es importante compartir con los artistas. Un formato uniforme de los nombres también facilitará que los colegas comprendan su significado.



Una escena bloqueada con activos etiquetados con nombres útiles, como "P_GenericWall_200x300cm": el prefijo P indica que es un Prefab, seguido del activo (una pared) y luego sus medidas.

© 2023 Unity Technologies

Machine Translated by Google

Para facilitar el acceso, también puede optar por utilizar texto flotante en el espacio 3D para
Identificar sus intenciones, permitiendo que un colega ejecute una compilación sin acceso de editor
y ver tu plan.

Finalmente, etiquetar sus activos no solo es útil para otros que miran su escena, sino que también es
También te resultará útil, ya que te ayudará a realizar un seguimiento de lo que estás trabajando.

Añade materiales a los activos en bloques para visualizar tu intención

Además de dar nombres descriptivos a los activos, también puedes aplicar materiales a
tus bloqueos para aclarar tu intención. Esta es una forma práctica de diferenciar
entre objetos interactivos y estáticos, espacio jugable y no jugable,
frágiles e irrompibles, etc.

Evite crear recursos artísticos detallados

> La etapa 3D es cuando necesitas empezar a pagar por tus conceptos y modelos... Para nosotros, fue la parte que más tiempo nos llevó: construir niveles y hacerlos realmente detallados. No quieres empezar a hacerlo hasta estar seguro de que tu nivel es exactamente como lo quieres.
>
> David Fenn, compositor y diseñador de sonido, codiseñador, productor y diseñador de niveles
> En la puerta de la muerte

El objetivo del white-boxing es probar ideas e iterarlas. Para evitar...
Al dedicar tiempo a actualizar activos que no están finalizados, su equipo no debería
crear activos 3D hasta que se hayan aprobado los diseños.

Una vez que tenga la aceptación de otros miembros relevantes del equipo y esté satisfecho con
En el diseño de su nivel, los artistas pueden comenzar a recorrer el nivel y agregar 3D.
activos para reemplazar las cajas blancas.

Pídele ayuda a un artista conceptual

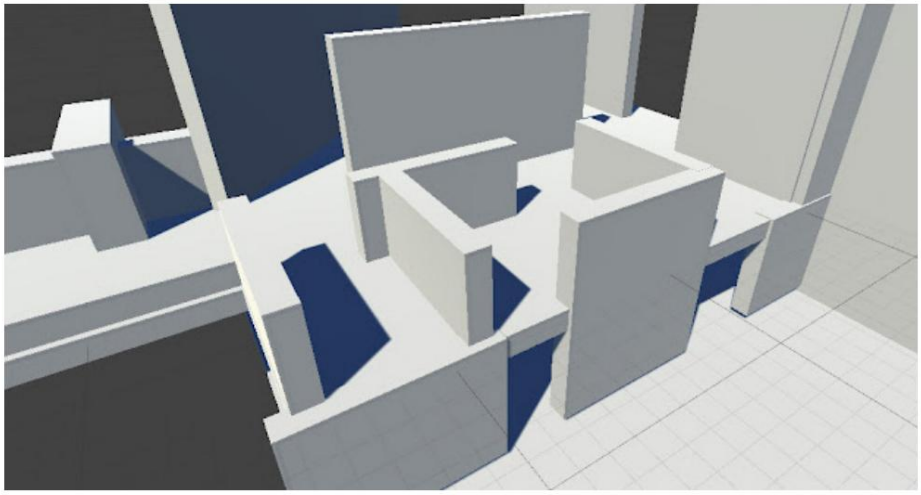
Los miembros del equipo podrían tener dificultades para visualizar el resultado final de una
escena de caja blanca. Un artista conceptual puede ayudarle a concretar su visión mediante la representación.
Una capa de pintura sobre la caja blanca. Esta es una representación 2D del nivel, por lo que hay
No es necesario crear recursos 3D. Normalmente, las pinturas se realizan tomando una
Toma una captura de pantalla del nivel y luego pintándola para mostrar el estilo y el estado de ánimo deseados.

También puedes compartir la escena real con un artista conceptual. Si tienes
documentado lo que representa cada uno de los bloques, será más fácil para el concepto
El artista se adherirá a su visión general.

© 2023 Unity Technologies

Machine Translated by Google

Siempre que el artista conceptual respete sus limitaciones y límites, una pintura adicional puede ser una buena forma de lograr la aceptación de su equipo.



Fuente de la imagen: Cómo usar el formato graybox y bloquear un videojuego 3D de de_crown, por FMPONE y Volcano – Bloqueo antes de pintar

Fuente de la imagen: Cómo usar el formato graybox y bloquear un videojuego 3D de de_crown, por FMPONE y Volcano – pintar sobre el blockout

Los activos ya preparados también pueden ayudar

Otra forma eficiente de amplificar su escena de caja blanca es colocar en ella recursos ya preparados.

La Tienda de Recursos de Unity ofrece una gran variedad de recursos listos para usar que te permiten visualizar tus ideas. Por ejemplo, el Paquete de Prototipos POLYGON. de Synty Studios te ayudará a comunicar las decisiones de diseño con tu equipo mediante las notas y marcadores incluidos. Consulta la Parte III de esta guía para obtener información detallada sobre cómo usar productos específicos de Unity Asset Store en el diseño de niveles.

© 2023 Unity Technologies

Machine Translated by Google

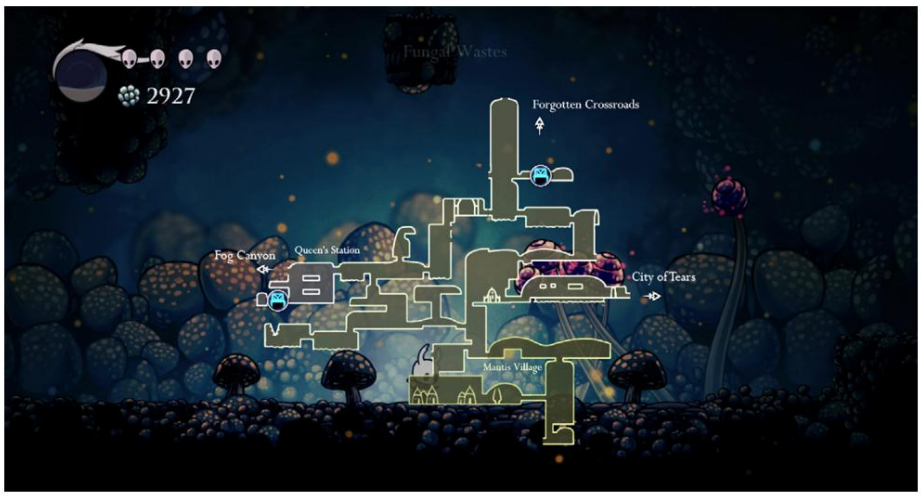
Paquete de prototipos POLYGON de Synty Studios, disponible en Unity Asset Store

Rutas de jugadores

El trazado de rutas describe los caminos que un jugador utiliza para alcanzar sus objetivos. Existen cuatro categorías de rutas en el desarrollo de juegos: críticas, de oro, secundarias y terciarias.

Ruta crítica

La ruta crítica es el camino más largo disponible para completar un juego. Piensa en ella como la ruta que debes seguir para probar todo el contenido que quieres publicar de la forma más eficiente.



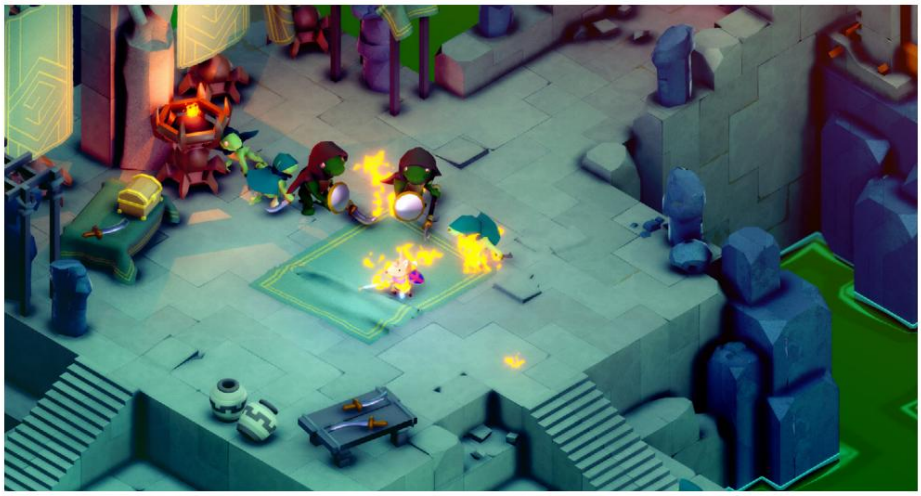
Hollow Knight, del equipo Cherry, es un popular juego 2D desarrollado con Unity. La ruta crítica sería la necesaria para completar el 112% del juego, es decir, la totalidad del mapa.

© 2023 Unity Technologies

Machine Translated by Google

Camino dorado

El camino dorado ofrece los mejores elementos de juego, recompensas y/o secretos.
Algunos juegos, como TUNIC, del equipo TUNIC, ofrecen múltiples finales, y el camino necesario para obtener el mejor final se considera el dorado.



TUNIC, del equipo TUNIC, es un juego de aventuras de acción isométrica que ofrece múltiples finales.

Dado que el camino dorado está pensado para ser el más atractivo y divertido para el jugador, asegúrese de probarlo con colegas para identificar puntos de frustración y confusión, así como aquellas partes que ayudan a que el contenido brille.

Normalmente, el camino dorado abarca la mayoría de los sistemas de juego, si no todos, para que el jugador experimente todo lo que ofrece. Sin embargo, no incluye todos los componentes.
Un juego con múltiples finales, por ejemplo, requeriría jugar cada uno para alcanzar el camino crítico, mientras que el camino dorado solo sería la vía más atractiva, interesante o divertida para llegar a un final.

Caminos secundarios y terciarios: misiones secundarias, secretos y atajos

¿Cuántos caminos necesitas? Idealmente, deberías ofrecer caminos para cada estilo de juego. Por ejemplo, Deus Ex: Human Revolution, de Eidos Montreal, se adapta a varios estilos de juego, como correr y disparar, sigilo y hackeo. Identifica los estilos de juego deseados para que los diseñadores puedan crear niveles con ellos en mente. Además, ofrece múltiples caminos para resolver un problema que dependan de cada estilo o de una combinación de ellos.

Hacer que un jugador retroceda por una zona ya explorada sin obtener prácticamente ninguna recompensa puede ser frustrante. Una mejor experiencia de juego consiste en asegurarse de que pueda volver rápidamente a la acción al finalizar un atajo o una misión secundaria.

Además, agregar contenido, incluso algo pequeño, al final de los caminos y espacios abiertos es una forma de recompensar al jugador por explorar y alentarlo a hacerlo a lo largo del juego.

© 2023 Unity Technologies

Machine Translated by Google

Pillars of Eternity, de Obsidian Entertainment, presenta un mundo encantado donde las decisiones que tomas, incluyendo qué Caminos a seguir, forja tu destino. La imagen de arriba muestra una selección de misiones secundarias, además de la principal.

La regla de tres para enseñar mecánicas a los jugadores

Dale tiempo a tus jugadores para que aprendan una nueva mecánica o sistema haciendo que lo repitan. varias veces durante una sesión de juego. La regla general es que el jugador debe realizar una nueva acción o serie de acciones al menos tres veces antes de que les resulte familiar con él. Espaciar nuevos sistemas y mecánicas para evitar sobrecargar a los jugadores.

Veamos un ejemplo de la regla de tres para introducir nuevas mecánicas en un Plataformas típicas:

1. Un jugador se encuentra con un solo enemigo por primera vez y descubre que...
   Puedes derrotarlo saltando sobre él.
2. Después de esa primera instancia, el jugador se encuentra con varios enemigos. que requieren que el jugador salte sobre cada uno de ellos una vez para derrotarlos. Esto consolida aún más la acción de saltar sobre la cabeza del enemigo para derrotarlo.
3. En la tercera secuencia, el jugador se encuentra con un grupo de enemigos espaciados aparte. El jugador puede saltar sobre la cabeza de cada enemigo para derrotarlos, pero esto El tiempo requiere más destreza.

Después de estas tres instancias, el jugador debería saber que puede derrotar a ambos jugadores individuales. y múltiples enemigos saltando sobre ellos una vez.

Divida una nueva mecánica o sistema en partes fáciles de aprender, como se explica en la sección “Identificar y comprender las diferentes mecánicas del juego”.

© 2023 Unity Technologies

Machine Translated by Google

Overcooked 2, de Team17, presenta la jugabilidad de forma progresiva, comenzando con acciones simples y pasando a recetas más elaboradas.

Subvirtiendo las expectativas

Ahora que has inculcado un patrón en la mente de tu jugador, ¡es hora de subvertirlo!
Los desafíos inesperados basados en nuevas mecánicas o la expansión de las existentes pueden aumentar el factor diversión y mantener a los jugadores interesados.

En el ejemplo anterior, el jugador aprendió que saltar sobre los enemigos una vez los derrota.

Ahora, el jugador se encuentra con un enemigo con casco. Instintivamente, salta sobre él, esperando vencerlo, pero solo el casco se destruye, mientras que el enemigo sobrevive. El jugador salta sobre el enemigo, ahora sin casco, y descubre que este segundo salto lo derrotará. Sus expectativas se ven trastocadas, pero sin frustración: el jugador aún puede confiar en el patrón familiar de saltar sobre un enemigo.

¡Marvel SNAP! de Second Dinner presenta a los jugadores cartas con habilidades que pueden crear emocionantes giros inesperados a lo largo del juego. Mira cómo... El equipo creó este éxito móvil en Unity, como se describe en su sesión GDC23.

Machine Translated by Google

Línea de visión y puntos ciegos

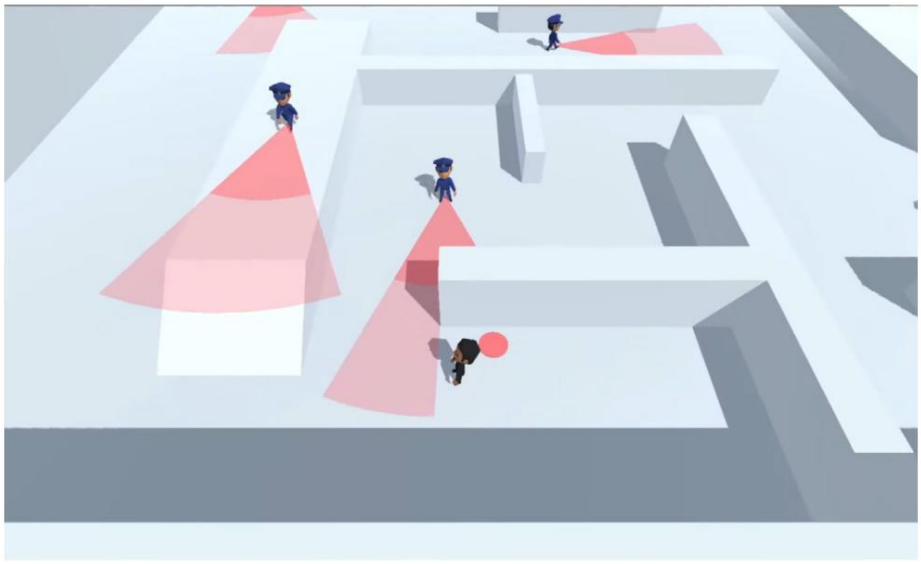


Imagen: Enemy Vision 2 – Detección de ruido, conos de dos niveles, reestructuración de código por Indie Marc

La línea de visión, También conocido como eje visual o línea de visión, es una línea imaginaria.
entre los ojos de un observador y un sujeto de interés o su dirección relativa.

La línea de visión se utiliza a menudo en los videojuegos para detectar enemigos, al jugador o
Para descartar objetos fuera de la línea de visión del jugador. Se puede identificar por un cono.
forma que consiste en un conjunto de líneas que parten de un punto de origen.

Por ejemplo, en los juegos FPS, los diseñadores de niveles prestan atención a las líneas de visión y
Puntos ciegos. Comprender dónde puede ver y ser visto el jugador a lo largo de un
El nivel es importante para crear desafíos para ellos, como dónde habrá tiroteos.
ocurrir y si hay o no cobertura suficiente para manejar alguna situación.
Alternativamente, puedes usar áreas de alta visibilidad para obligar al jugador a encontrar un
ruta más segura.

En contraste con la línea de visión se encuentran los puntos ciegos, áreas que no están inmediatamente visibles.
visible para el jugador o la IA. Usa puntos ciegos para generar enemigos fuera del área.
la visión del jugador, coloca secretos y crea intensos momentos de sigilo.

Puedes crear líneas de visión y sistemas similares con raycasts y física en
Unidad, que se explica en la Parte II.

Llamar la atención del jugador

En muchos juegos, querrás dirigir y alentar al jugador a explorar algunos
espacios mientras los animamos a evitar otros. Veamos técnicas que puedes
Úselo para dirigir la atención y el movimiento de un jugador.

© 2023 Unity Technologies

Machine Translated by Google

En INSIDE, el juego de aventuras independiente de Playdead, se utilizan luces durante todo el juego para atraer la atención del jugador hacia los caminos y puntos de interés en la pantalla.

Iluminación y espacio

La iluminación de un espacio de juego contribuye en gran medida a orientar la actitud del jugador.
movimiento y creación de la atmósfera de una escena.

Utilice la iluminación para enfatizar un camino particular, como una luz al final del túnel,
para dirigir al jugador hacia él. La falta de luz también es efectiva: una luz tenue u oscura
El espacio puede generar miedo en el jugador, manteniéndolo alejado de los enemigos acechantes.

Bloqueadores físicos

Los bloqueadores físicos son colisiones que bloquean el movimiento del jugador. Usa bloqueos físicos.
Bloqueadores para ayudar a dirigir al jugador y mantenerlo en el espacio jugable. Marcar o
Indicar claramente un bloqueador, por ejemplo, añadiendo una señal visual a un objeto invisible.
colisiones que marcan los límites del espacio jugable.

© 2023 Unity Technologies

Machine Translated by Google

Algunas formas que puedes utilizar para limitar a tu reproductor a un espacio jugable incluyen:

— Un espacio jugable suspendido rodeado o encerrado por un piso de “muerte instantánea” que mata al jugador si entra en contacto con él.

— Un límite en los recursos disponibles para un jugador que le impide explorar un espacio demasiado lejos de un punto anclado

— Un límite invisible que, cuando se traspasa, activa una notificación o mensaje para avisar al jugador que tiene un tiempo limitado más allá del límite antes de ser asesinado o devuelto (normalmente se usa en juegos de batalla real)

Por supuesto, hay muchas maneras de contener a tus jugadores dentro del espacio jugable, así que explora y explota los sistemas que mejor se adapten a tu juego.

Oración a los dioses, De No Matter Studios, es un juego de aventuras de mundo abierto donde luchas contra jefes y te conviertes en un héroe solitario enviado al borde de una tierra gélida y agonizante para explorar y resolver los secretos de un invierno eterno. A lo largo del juego, los bloqueadores físicos y las señales guían al jugador a medida que aparece en el mundo.

Señalización

La señalización se refiere a pistas, señales u otro contenido que ayuda al jugador a comprender, navegar y progresar en cada nivel. Mantiene al jugador informado sobre dónde ir y proporciona pistas e indicaciones a las que recurrir si se pierde.

Así como el mundo real está lleno de señalización, desde mapas y señales de tráfico hasta carteles de "empujar" o "tirar" en una puerta, un juego debe proporcionar suficientes pistas y mensajes para ayudar a los jugadores a comprender el mundo en el que se encuentran. Utilice tanto una señalización sutil como algo tan obvio como un punto de referencia que se pueda ver y reconocer fácilmente a distancia.

Los colores se pueden usar para señalizar cuando su significado es fácilmente reconocible por una audiencia global; por ejemplo, rojo para detenerse/peligro, verde para avanzar, etc. Muchos jugadores sabrán que un barril rojo brillante puede indicar peligro, como una explosión inminente.

© 2023 Unity Technologies

Machine Translated by Google

Sonido

El sonido es otra excelente manera de captar la atención del jugador. Ya sean disparos a lo lejos, una voz que lo llama hacia adelante o un ruido chirriante y aterrador en la oscuridad, el sonido enriquece la experiencia del jugador y ayuda a reforzar las mecánicas y acciones del juego.

El juego Subnautica hace un excelente trabajo usando el sonido para ayudar al jugador a comprender su estado. Por ejemplo, cuando el jugador tiene poco oxígeno, la música se vuelve más intensa y turbia, y se ven y se oyen burbujas saliendo de su boca antes de que todo se apague.

Utiliza pistas auditivas y visuales para adaptarse a jugadores sordos o con discapacidad auditiva. En Subnautica, las pistas visuales, como los mensajes de texto de la interfaz de usuario, acompañan a las pistas auditivas para indicar el estado del jugador.

En Subnautica, Por Unknown Worlds Entertainment, los jugadores sobreviven construyendo hábitats, fabricando herramientas y adentrándose en el mundo submarino. En esta imagen, el jugador está a punto de quedarse sin oxígeno, como lo indican las estadísticas de salud, el texto de la interfaz y los efectos de sonido.

Puntos de aparición

Generar al jugador en tu mundo marca el tono de la experiencia y puede guiarlo en la dirección deseada. Es recomendable generar al jugador mirando en la dirección deseada, para que no tenga que girar ni regresar por un camino que ya ha recorrido.

Aprovecha el momento de aparición para enfocar la cámara hacia elementos importantes. Esto indica su importancia antes de devolverle el control al jugador.

Puntos de guardado y puntos de control

Los puntos de guardado y los puntos de control, generalmente colocados por el diseñador de niveles, ayudan a mantener a los jugadores enganchados a la acción. Guardar el progreso de un jugador lo anima a probar cosas sin miedo a tener que empezar de cero.

© 2023 Unity Technologies

Machine Translated by Google

En Hollow Knight del Equipo Cherry, los puntos de guardado y el banco son parte del mundo y de la narración.

Puntos de guardado

En algunos juegos, un punto de guardado puede ser estático, lo que requiere que el jugador lo busque manualmente para guardar la partida. Los puntos de partida estáticos permiten al diseñador controlar qué se guarda, cuándo y dónde, y son útiles cuando guardar la partida en un lugar determinado resulta demasiado complejo para el sistema.

En otros juegos, los jugadores pueden guardar prácticamente en cualquier momento del juego o con algunas restricciones. Normalmente, esto requiere más esfuerzo y un código de juego robusto, ya que recargar una partida guardada en cualquier lugar tras completar una acción puede ser complicado.

Muchos juegos usan una combinación de ambos tipos de puntos de guardado. Por ejemplo, un juego podría permitir guardar en cualquier lugar, pero todos los enemigos se reinician y apareces en una ubicación estática establecida por el diseñador, en lugar de donde guardaste la partida. Este enfoque combinado puede funcionar bien en juegos complejos, permitiendo a los jugadores guardar en cualquier momento y manteniendo un sistema de guardado lo suficientemente simple como para minimizar posibles errores y problemas.

Puntos de control

Los puntos de control se colocan a lo largo de un nivel, se guardan en la memoria temporalmente y solo son accesibles mientras el jugador progresa en una misión o nivel.

Esta técnica permite al diseñador establecer partidas guardadas temporales a lo largo de una misión larga, dividiéndola en secciones más pequeñas y facilitando el progreso del jugador. Los puntos de control también son útiles cuando hay varios desafíos en una secuencia o la misión es demasiado compleja para recargarla en cualquier momento.

Evite el bloqueo suave

Asegúrate de probar todos los puntos de control del juego para evitar que el jugador se bloquee. Un bloqueo se produce cuando el jugador ya no puede avanzar ni retroceder en el juego, lo que a menudo lo obliga a cargar una partida guardada anterior o reiniciarla por completo.

© 2023 Unity Technologies

Machine Translated by Google

Algunos ejemplos de un bloqueo suave incluyen:

— Cuando un jugador se ve obligado a recargarse para colisionar y no puede escapar.

— Activar un punto de control en un punto del juego donde el jugador se encuentra
No están preparados para seguir avanzando, pero al mismo tiempo no pueden retroceder.

— Recargar una partida guardada y morir inevitablemente

Los juegos con múltiples ranuras de guardado automático permiten al jugador guardar manualmente en diferentes puntos del juego, lo que le da la posibilidad de revertir su progreso y evitar el bloqueo suave.

Fuerza de hermanos Free Lives es un juego de plataformas de correr y disparar con desplazamiento lateral que utiliza inteligentemente puntos de guardado y puntos de control para segmentar el juego mientras mantiene a los jugadores en la acción.

Ubicación consciente de los puntos de guardado y puntos de control

Ten cuidado de dónde colocas los puntos de guardado y de control a lo largo del juego para evitar frustrar al jugador. Por ejemplo, evita colocar un punto de guardado para una misión difícil antes de una cinemática larga que no se pueda saltar; si cada vez que el jugador falla un nivel se ve obligado a ver la cinemática, es probable que se frustre rápidamente. Intenta colocar un punto de guardado o de control después de cinemáticas largas para que el jugador pueda volver a la acción de inmediato. Como alternativa, permite que el jugador se salte una cinemática.

Esto también aplica a las secciones de juego con desafíos secuenciales difíciles. Por ejemplo, si un jugador completa una sección de plataformas difícil y luego se enfrenta a un jefe importante, podría ser más beneficioso para la jugabilidad separar estos desafíos colocando un punto de guardado o un punto de control entre ellos.

Diseño de nivel procedimental

La generación procedural es una herramienta poderosa que facilita la creación de grandes cantidades de contenido en los juegos. Puedes usarla para ahorrar tiempo de desarrollo, reducir el tamaño del proyecto o generar diseños aleatorios para una jugabilidad menos predecible.

© 2023 Unity Technologies

Machine Translated by Google

En Timberborn, de Mechanistry, el terreno generado mediante procedimientos ofrece a los jugadores un rompecabezas y un patio de juegos, como este "pueblo inicial", para construir sus ciudades.

Existen varias maneras de incorporar la generación procedural al desarrollo de videojuegos, y la tecnología está en constante evolución. Analiza qué tipo de generación procedural se adapta mejor al tipo de proyecto que estás creando. Para ver ejemplos del método en Unity, consulta esta entrada del blog. sobre patrones procedimentales con mapas de mosaicos.

La generación procedural produce resultados basados en algoritmos y el contenido proporcionado. El contenido suele ser creado por el equipo de desarrollo, pero también puede ser generado por el usuario mediante herramientas de creación de niveles disponibles para el jugador o la comunidad. Los datos SIG también pueden utilizarse para contenido procedural, lo que permite al creador obtener datos de la geografía real e importarlos a un juego como un nivel jugable.

Reglas de generación de procedimientos

Aunque la generación procedimental es excelente para añadir aleatoriedad a tus niveles, también puede generar diseños confusos y frustrantes. Funciona mejor con reglas fáciles de entender y usar. Unas buenas directrices ayudarán al jugador a conseguir el resultado deseado manteniendo la aleatoriedad. Sin embargo, para realizar cambios significativos, es necesario comprender la relación entre los elementos generados procedimentalmente.

Puede ser útil construir el contenido generado procedimentalmente en torno a estructuras únicas o reconocibles que sirvan de guía al jugador al explorar el contenido generado. Este tipo de puntos de referencia actúan como herramientas de navegación visual y pueden ayudar a romper la monotonía generada por el sistema.

© 2023 Unity Technologies

Machine Translated by Google

Pruebas automatizadas

Dado que la generación procedimental tiende a ofrecer una cantidad significativa de resultados, vale la pena crear pruebas automatizadas para ahorrar tiempo y esfuerzo.

Configure sus pruebas automatizadas para ejecutar la generación de niveles procedimentales a través de cientos de iteraciones. Esto sirve para comprobar si las reglas utilizadas funcionan correctamente o si es necesario modificarlas. Dado que el ordenador ejecutará las pruebas sin intervención del usuario, debería ser mucho más rápido generar muchas iteraciones de su nivel en poco tiempo. Una vez que tenga los datos a mano, podrá realizar los ajustes necesarios a sus reglas.

Prueba, prueba y vuelve a probar

Es fundamental probar su contenido (por eso se menciona varias veces a lo largo de esta guía).

¿Cómo se desarrollan tus niveles? ¿Cuáles son, y dónde están, los puntos fuertes y débiles de tu diseño, y cómo puedes aprovecharlos y mejorarlos? Jugar con tu contenido (y animar a tu equipo a que lo haga) te ayudará a identificar problemas de usabilidad, errores y bloqueos leves antes de que lleguen al grupo de control de calidad.

Al diseñar tu contenido, piensa en qué herramientas podrían hacer que las pruebas sean más eficientes. ¿Necesitas saber a qué distancia está tu personaje de otros jugadores? Podría ser útil crear un script para mostrar la distancia. ¿Quieres saber dónde falla el jugador con más frecuencia en un mapa? Registra dónde falla el jugador y crea un mapa de calor.

Si necesitas ayuda para probar aspectos específicos del juego o necesitas comentarios sobre los niveles, el equipo de pruebas es una excelente fuente de información. También estarán en una buena posición para informarte sobre el tipo de pruebas que realizan y las herramientas que podrían optimizar su trabajo.

Valheim, del estudio Iron Gate, es un juego con mundos generados proceduralmente. Los niveles siempre tienen el mismo tamaño, y el jugador comienza en el centro.

© 2023 Unity Technologies

Machine Translated by Google

PARTE II : INTRODUCCIÓN TOUNITYFOR DISEÑADORES DE NIVELES

No siempre se espera que los diseñadores de niveles tengan experiencia con Unity.
Sin embargo, debido a su formación y experiencia, deberían tener las habilidades lógicas
necesario para ayudarles a empezar con el software.

Esta sección proporciona una introducción al trabajo en Unity, incluida la instalación de Unity,
Comprender la interfaz del Editor, comprender los elementos fundamentales para
cualquier escena que crees en Unity, organizando tus recursos y proyectos, y más. Es
Se recomienda que lea esta sección antes de la tercera y última sección que
Cubre conjuntos de herramientas de diseño de niveles específicos de Unity y Unity Asset Store.

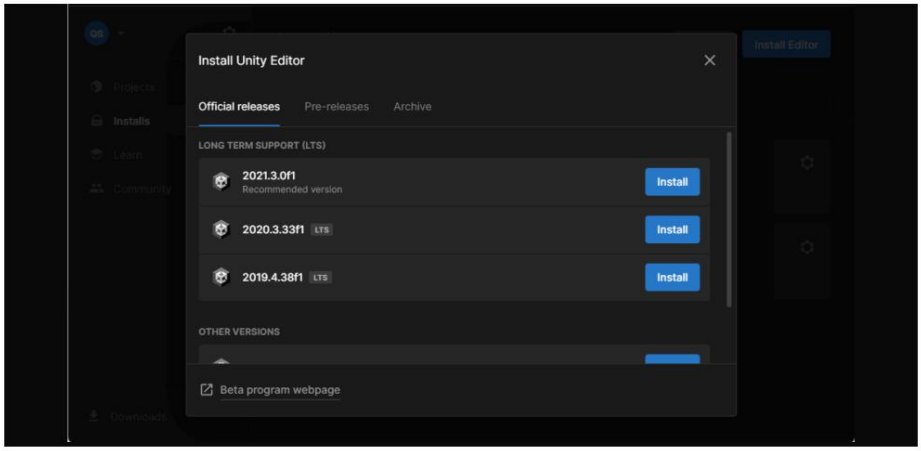
Los materiales de aprendizaje más completos para usuarios principiantes de Unity son
disponible en Unity Learn. El Camino Los cursos creados por expertos de Unity son los
Los mejores recursos para empezar. También recomendamos " Crear con código". serie,
La aventura de Ruby para principiantes en 2D y Haunted Jaunt de John Lemon para 3D
Principiantes. Para tutoriales más cortos, prueba los proyectos del Kit de creación para un juego de rol. Rompecabezas o
FPS juego, todos ellos diseñados para completarse en una o dos horas.

Instalar Unity

Hay dos versiones de Unity disponibles para descargar: Soporte a largo plazo (LTS)
o Tech Stream. La edición LTS es la versión predeterminada que incluye las funciones.
y mejoras realizadas a lo largo del año en una única instalación que proporciona
Máxima estabilidad. El Tech Stream ofrece acceso anticipado a nuevas funciones y
Se recomienda principalmente para la preproducción, el descubrimiento y la creación de prototipos.
Fases de desarrollo. Puedes leer más sobre las versiones de Unity aquí.

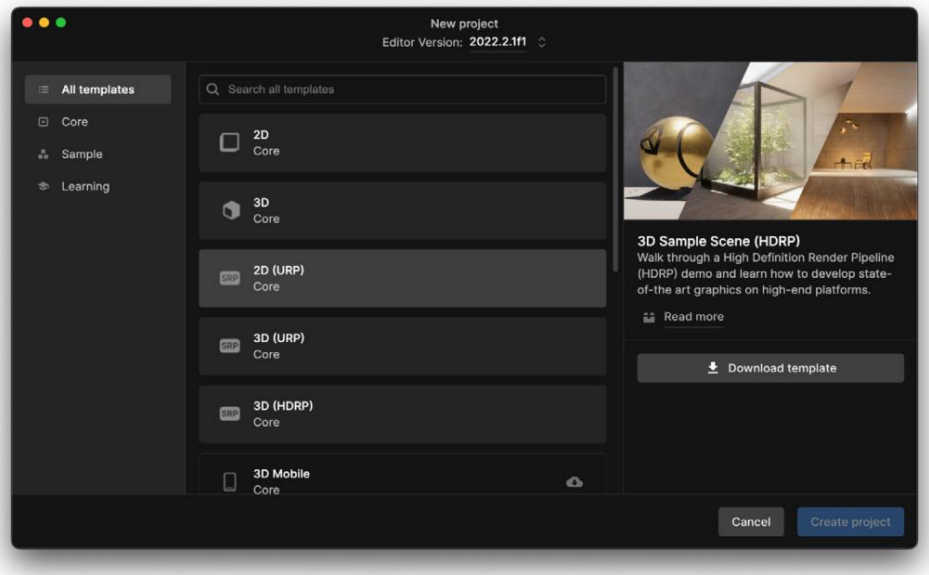
© 2023 Unity Technologies

Machine Translated by Google



Puede administrar sus instalaciones, proyectos y módulos de Unity en Unity Hub.

Se recomienda que instales la última versión de Unity LTS, lo que puedes hacer usando el iniciador de Unity Hub, disponible en el sitio web de descargas.



La página de Proyectos en Unity Hub

En Unity Hub, la página Proyectos muestra tus proyectos de Unity. Úsala para crear un proyecto nuevo, administrar tus proyectos existentes o abrir uno en el Editor de Unity.

Al seleccionar "Nuevo proyecto", verá varias opciones que podrían cambiar según la versión de Unity que tenga instalada. Si instala Unity 2022 LTS, aparecerán las siguientes plantillas vacías en la lista:

— 2D: Configurado para el desarrollo de juegos 2D que utilizan una versión anterior de Canal de renderizado integrado de Unity y viene con paquetes 2D precargados

— 3D: Configurado para un proyecto 3D con el canal de renderizado integrado

© 2023 Unity Technologies

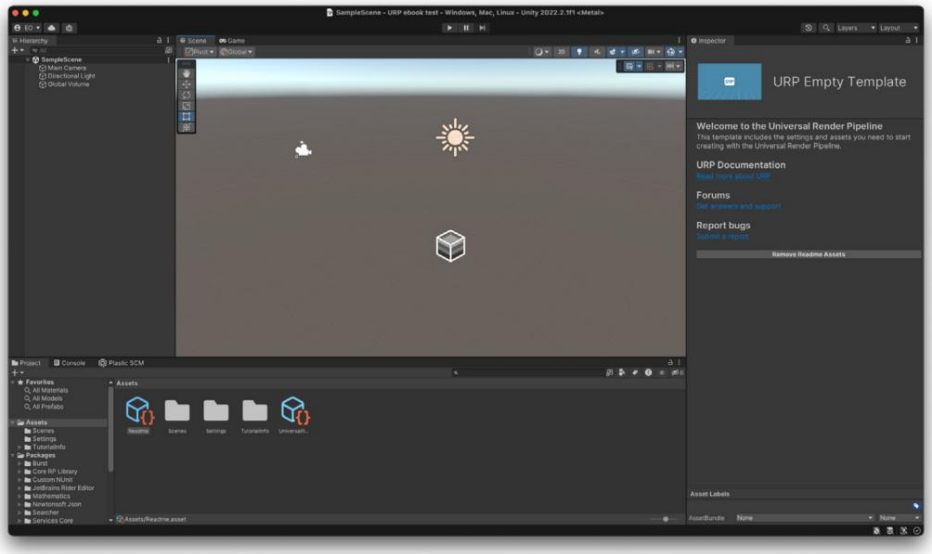
Machine Translated by Google

— 2D (URP): una plantilla 2D más nueva que utiliza Universal Render Pipeline (URP), que permite funciones gráficas 2D actualizadas, como luces y sombras 2D

— 3D (URP): plantilla 3D que utiliza URP y viene preconfigurada para soportar Rendimiento, amplio soporte de plataformas y personalización de gráficos

— 3D (HDRP): incluye configuraciones para admitir proyectos que apuntan a plataformas de alta gama y requieren funciones disponibles en el renderizado de alta definición. Tubería (HDRP)

En el Hub también se encuentran disponibles plantillas adicionales específicas para plataformas de destino, como por ejemplo para juegos móviles en 3D.



Un proyecto vacío puede resultar desalentador si no estás familiarizado con las numerosas características que ofrece Unity, pero no temas: puedes comenzar rápidamente a crear tus propios activos o importar los que ya estén listos para usar desde Unity Asset Store.

El editor de Unity

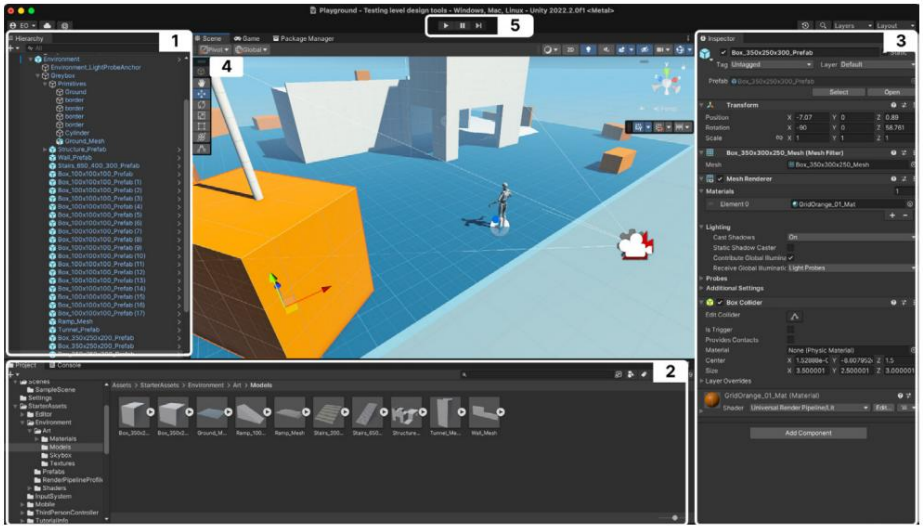
Puedes reorganizar, redimensionar, ocultar o mostrar ventanas en el Editor de Unity para adaptarlas a tus necesidades. Puedes guardar tus diseños personalizados. para cambiar rápidamente entre ellos.

En la imagen a continuación, puedes ver las vistas predeterminadas en el Editor:

1. Ventana de jerarquía: Muestra todos los GameObject en una escena, como modelos, cámaras o prefabricados.
2. Ventana del proyecto: Muestra todos los activos de un proyecto, como scripts, modelos 3D, prefabricados o texturas.
3. Ventana del inspector: Muestra propiedades configurables de un activo o GameObject y los scripts relacionados con él
4. Vista de la escena: Muestra la escena en desarrollo.
5. Vista del juego Botones Reproducir, Pausa y Paso: Disponibles a través de la barra de herramientas, Estos botones activan la vista del Juego, que muestra el juego ejecutándose en el Editor y te permite jugarlo, probarlo e iterarlo.

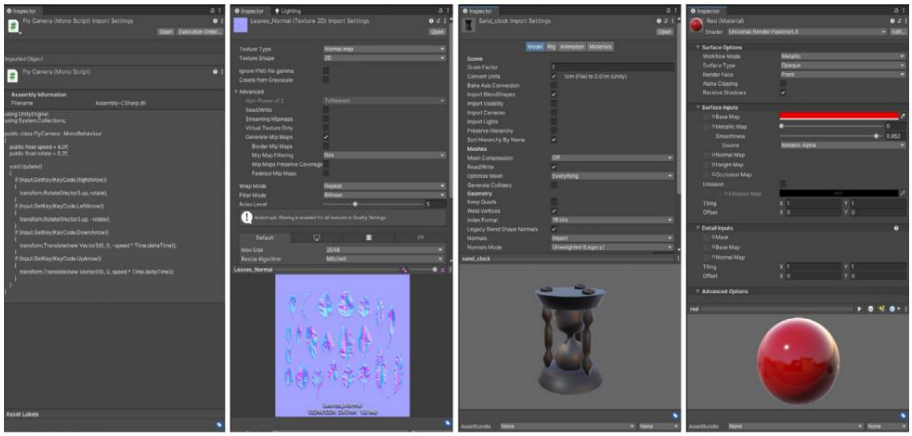
© 2023 Unity Technologies

Machine Translated by Google



El Editor con las ventanas Jerarquía, Inspector y Proyecto, y la vista Escena

Por favor, consulte la interfaz de Unity Sección en la documentación para obtener información detallada Explicación del diseño y la funcionalidad de cada ventana y vista del editor.



Diferentes activos en la vista del Proyecto y sus configuraciones en la ventana del Inspector

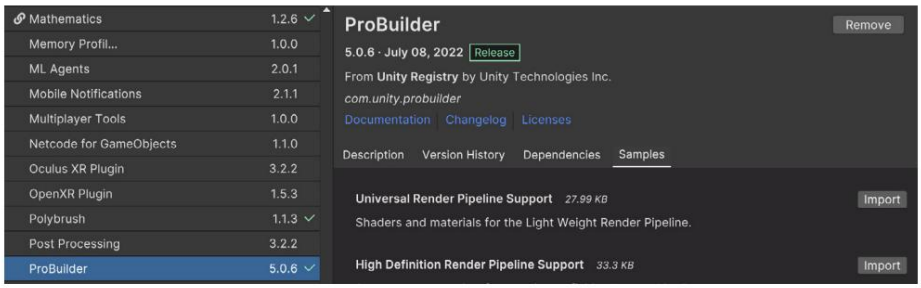
Vista de jerarquía: en este ejemplo, los GameObjects que son prefabricados reutilizables se indican con un icono azul. El orden de La lista de GameObjects en la vista de Jerarquía no tiene ningún impacto en el proyecto; una excepción es cuando trabajas con GameObjects dentro de un lienzo de UI de Unity.

© 2023 Unity Technologies

Machine Translated by Google

Administrador de paquetes

Muchas funciones de Unity no están precargadas con un nuevo proyecto, sino que están disponibles como paquetes modulares en el Administrador de paquetes. Desde Ventana > Administrador de paquetes en el Editor. En el Administrador de paquetes, puede ver las versiones disponibles de cada paquete e instalar, eliminar, deshabilitar o actualizar paquetes para cada proyecto.



El Registro de Unity en el Administrador de paquetes

En la lista desplegable en la parte superior de la ventana del Administrador de paquetes, encontrará el Registro de Unity, el menú predeterminado que enumera todos los paquetes oficiales de Unity disponibles. Los paquetes ya instalados en un proyecto se listan en En el proyecto. Las compras en Unity Asset Store se listan en Mis activos, y las funciones y paquetes preinstalados se listan en Integrados.

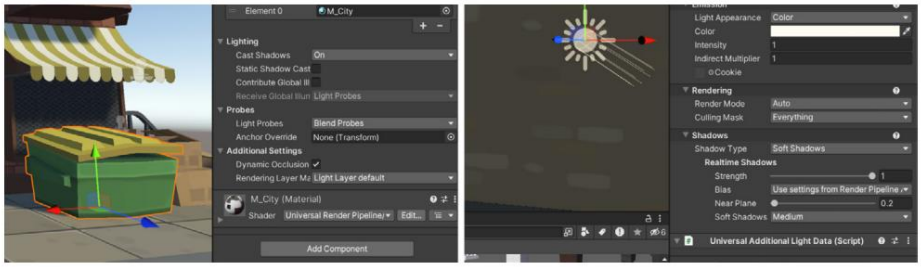
Lea sobre todos los paquetes disponibles en Unity 2022 LTS aquí.

Objetos del juego

Cada objeto en una escena de Unity comienza como un GameObject, Como personajes, objetos coleccionables, luces, cámaras y efectos especiales. Un GameObject, por sí solo, es un contenedor vacío. Antes de que pueda convertirse en algo y realizar alguna función en un juego, es necesario asignarle funcionalidad mediante la adición de componentes. Los componentes, a su vez, proporcionan un conjunto de propiedades editables para implementar su funcionalidad.

Un GameObject siempre tiene una Transformación componente adjunto (para representar posición, escala y orientación en el espacio 3D) que no se puede quitar.

Agregue componentes mediante el menú "Agregar componente" del Editor. En la lista desplegable, seleccione entre los componentes predefinidos o defina la funcionalidad de sus propios componentes con scripts.



Dos tipos diferentes de GameObjects: un modelo 3D y una luz direccional; observe cómo el componente Transform y los gizmos son los mismos para ambos, pero el resto de los componentes difieren.

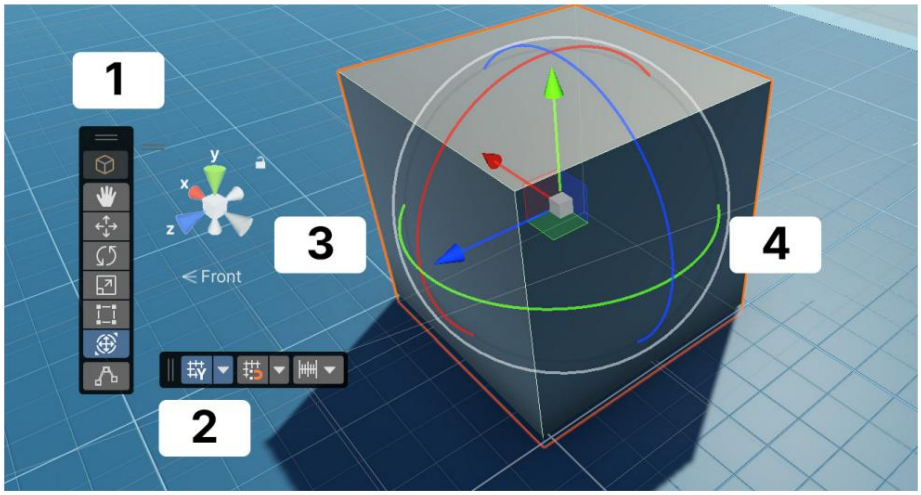
© 2023 Unity Technologies

Machine Translated by Google

Manipulación de GameObjects en la vista de escena

Unity proporciona una cuadrícula visual en la vista de escena para ayudarlo a alinear GameObjects
chasquido (moviéndolos) a la cuadrícula.

Navegar por una escena y manipular objetos será una gran parte del trabajo para
Diseñadores de niveles que usan Unity. Considere familiarizarse con los atajos. Para el
herramientas de transformación y personalizarlas si es necesario. Además, el panel se superpone
En la vista de escena se muestran las herramientas que también se pueden configurar a través de las superposiciones.
menú (consulte el menú de accesos directos para encontrar la tecla para acceder en su sistema operativo).



Paneles superpuestos predeterminados en la vista de escena

Las opciones disponibles en Herramientas El panel puede variar según el seleccionado
objeto, pero la siguiente tabla muestra las herramientas principales y los accesos directos predeterminados
de uso frecuente.

Descripción de la herramienta		Ventanas	Impermeable
	Mueva la cámara en la escena vista	Q o cambio	Q o cambio
	Acercar y alejar con el ratón desplazamiento o atajos	Alt + clic derecho Opción + clic derecho	
	Orbita la cámara alrededor de la vista del centro de la escena	Alt + clic izquierdo Opción + clic izquierdo	
	Seleccionar elementos y moverlos W		O
	Cambiar el ángulo de rotación de objetos	mi	mi
	Escalar objetos seleccionados	R	R
	Transforma la escala utilizando la cuadro límite del objeto (también conocida como herramienta Rectángulo)	T	T

© 2023 Unity Technologies

Machine Translated by Google

	Transforme la posición, rotación o escala del objeto usando el gizmo como se muestra en la imagen (4)	Y	Y
	Esta herramienta aparece cuando se adjunta un componente Collider a un GameObject. Permite manipular los límites del hitbox o del colisionador. objeto.	-	-

La cuadrícula y el ajuste Las herramientas ayudan a alinear GameObjects ajustándolos (moviéndolos) a la ubicación de la cuadrícula más cercana.

Herramienta	Descripción
	Activa y desactiva la cuadrícula, cambia el eje y la opacidad de la cuadrícula, alinea el plano de la cuadrícula con el objeto seleccionado (Al controlador) o la posición 0 del eje respectivo en la escena (Al origen).
	Active o desactive el ajuste de la cuadrícula y la lista desplegable le permitirá configurar los intervalos en los que puede mover el objeto (de manera predeterminada, 1 unidad en todos los ejes, pero puede personalizarlo por eje). También puedes alinear el GameObject seleccionado a la cuadrícula en todos los eje o seleccionados.
	El ajuste incremental ofrece configuraciones para el ajuste cuando mueves, giras o escalas un GameObject mediante Control en Windows o Comando en Mac; en la lista desplegable, puedes configurar los intervalos para ello.
Vértice chasquido	De forma predeterminada, cada vez que manipulas un GameObject lo haces desde el centro del objeto o el punto pivote (verifica la posición del controlador) . (Alternar), pero si quieres hacerlo desde un vértice de la malla, mantén presionada la tecla V y mueve el ratón para seleccionar el vértice. Esto es especialmente útil para, por ejemplo, manipular una pared desde una de sus esquinas y alinearla perfectamente con otras estructuras. Encuentra más detalles sobre el posicionamiento de GameObjects en la documentación. Un consejo que los diseñadores de palanca compartieron con nosotros es intentar configurar el pivote de un modelo 3D en las herramientas DCC en la posición 0,0,0, de modo que siempre pueda esperar poder arrastrarlo desde la esquina inferior cuando el modo de manija esté configurado en el punto de pivote.

El artilugio de orientación tiene un brazo cónico en cada lado del cubo. Los brazos en el

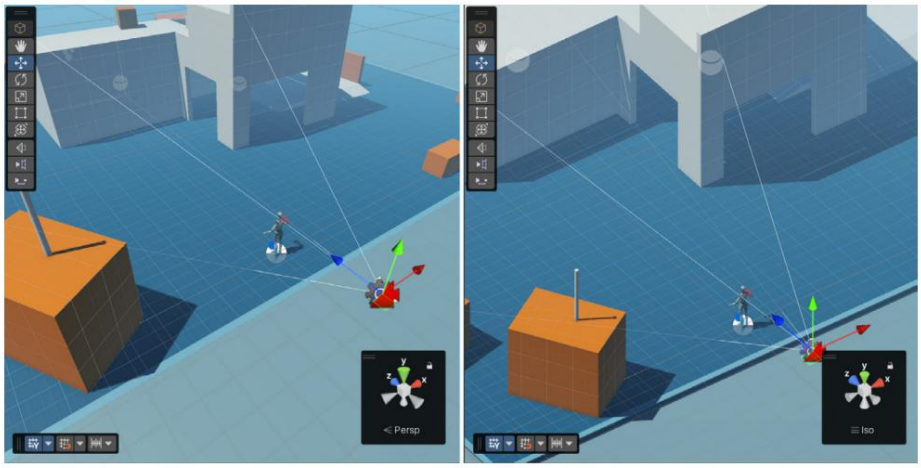
Las vistas frontales están etiquetadas como X, Y y Z. Esto muestra la orientación actual de la cámara de la vista de escena y permite modificar rápidamente el ángulo de visión y el modo de proyección al hacer clic en estos conos. Al hacer clic en el cubo blanco en el centro del Gizmo o en el texto de la parte inferior, se puede alternar entre las cámaras Perspectiva y Ortográfica (esta última a veces se denomina cámara isométrica o 2D). El texto debajo del Gizmo indica la vista actual. El icono del candado activa o desactiva la rotación de la vista de la cámara. Esto puede ser útil cuando...

El juego tiene un ángulo de cámara fijo con el que deseas trabajar la mayor parte del tiempo.

© 2023 Unity Technologies

Machine Translated by Google

Los artilugios de escena para manipular un GameObject en un espacio 3D siempre seguirán el siguiente código de color: rojo para manipular el eje X, azul para el eje Z o profundidad y verde para el eje Y.



Una vista de cámara en perspectiva a la izquierda y ortográfica a la derecha.

Creación de GameObjects

Puedes crear un GameObject de dos maneras: desde la barra superior Menú > GameObject, o arrastrando y soltando activos directamente en la ventana Jerarquía o en la vista Escena.

Unity utiliza el concepto de jerarquías padre-hijo, o parenting, para agrupar GameObjects. Aprende sobre parenting, organización de GameObjects, creación de GameObjects hijos y más en la página de la ventana Jerarquía. en la documentación de Unity.

No todos los recursos se pueden convertir automáticamente en un GameObject, pero los tipos más comunes sí. Si añades un archivo FBX (modelo 3D) a la escena, aparecerá en la jerarquía un nuevo GameObject con los componentes necesarios para visualizar el modelo.

GameObjects estáticos y dinámicos

Los GameObjects que no se mueven durante el tiempo de ejecución, como accesorios o modelos, se conocen como GameObjects estáticos. Se pueden marcar como estáticos marcando la casilla "Estático" a la derecha del campo "Nombre del GameObject". Los GameObjects dinámicos son aquellos que se mueven durante la ejecución.

Muchos sistemas en Unity pueden precalcular información sobre GameObjects estáticos en el Editor. Esto significa que Unity puede ahorrar cálculos en tiempo de ejecución y ayudar a mejorar el rendimiento. Al cambiar la configuración estática de un GameObject padre, se le solicitará que decida si desea que todos sus hijos tengan la misma configuración estática actualizada, lo que puede ahorrarle tiempo considerable si tiene un proyecto con cientos de propiedades anidadas.

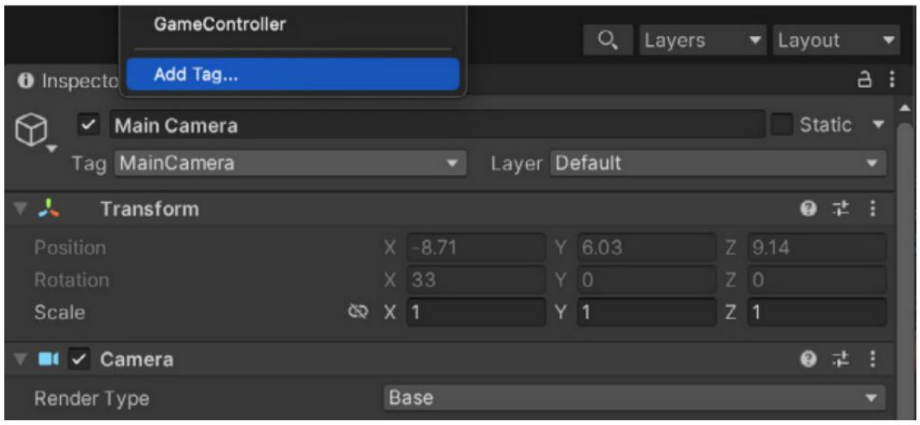
© 2023 Unity Technologies

Machine Translated by Google

GameObjects activos/inactivos

Puedes marcar un GameObject como inactivo para eliminarlo temporalmente de la vista de escena. Un ejemplo común es tener GameObjects inactivos en una escena que se activan durante el juego cuando el jugador llega a cierto punto.

Los componentes asociados a GameObjects inactivos también se desactivan. Al desactivar un GameObject principal, también se desactivan todos sus GameObjects secundarios.



Puede asignar etiquetas desde el menú desplegable y agregar nuevas etiquetas en Agregar etiqueta.

Etiquetas

Una etiqueta es una palabra de referencia que puedes asignar a uno o más GameObjects. Por ejemplo, puedes agregar etiquetas como "Jugador" para personajes controlados por el jugador, una etiqueta como "Enemigo" para personajes no controlados por el jugador, una etiqueta como "Coleccionable" para objetos que el jugador puede coleccionar, etc.

Las etiquetas ayudan a identificar GameObjects al crear scripts. Usar etiquetas es una forma más óptima de referenciar GameObjects que por su nombre, ya que este puede cambiar durante el desarrollo. Las etiquetas son útiles para la lógica de detección de colisiones. Por ejemplo, si el GameObject que colisionó tiene la etiqueta "Enemy", podría ser útil ejecutar alguna lógica con ese GameObject, como deshabilitarlo. Los equipos de desarrollo deben llegar a un acuerdo desde el principio de la producción del juego sobre cómo etiquetar los objetos.

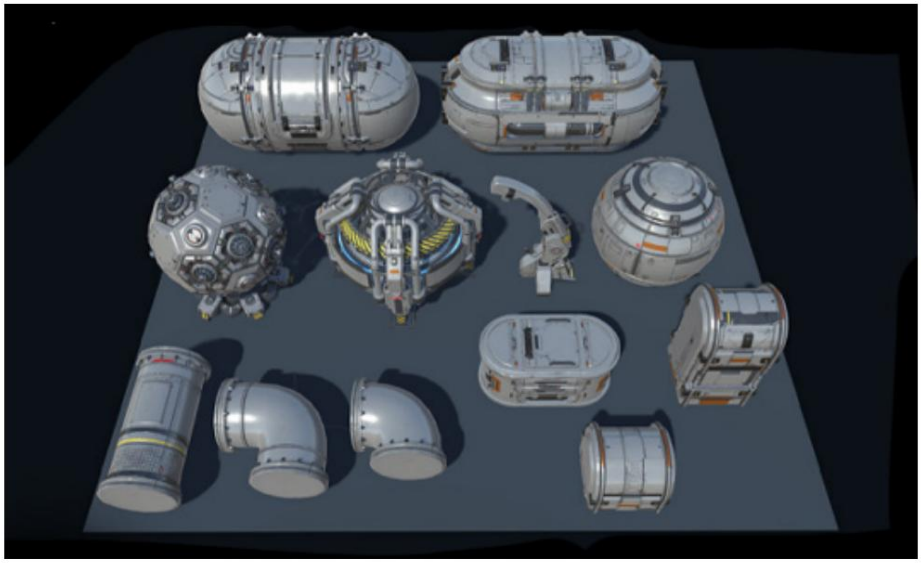
Prefabricados: GameObjects reutilizables

El prefabricado El sistema es una herramienta para completar un nivel de manera confiable y eficiente, lo que lo convierte en una de las herramientas más importantes para usar como diseñador de niveles.

Un GameObject recién creado en la vista de Escena solo pertenece a esa escena. Puedes duplicar el objeto, pero si necesitas modificarlo posteriormente, debes hacerlo manualmente en cada duplicado. Claramente, esta no es una forma viable de crear un juego donde muchos elementos se repiten con frecuencia en las escenas.

© 2023 Unity Technologies

Machine Translated by Google

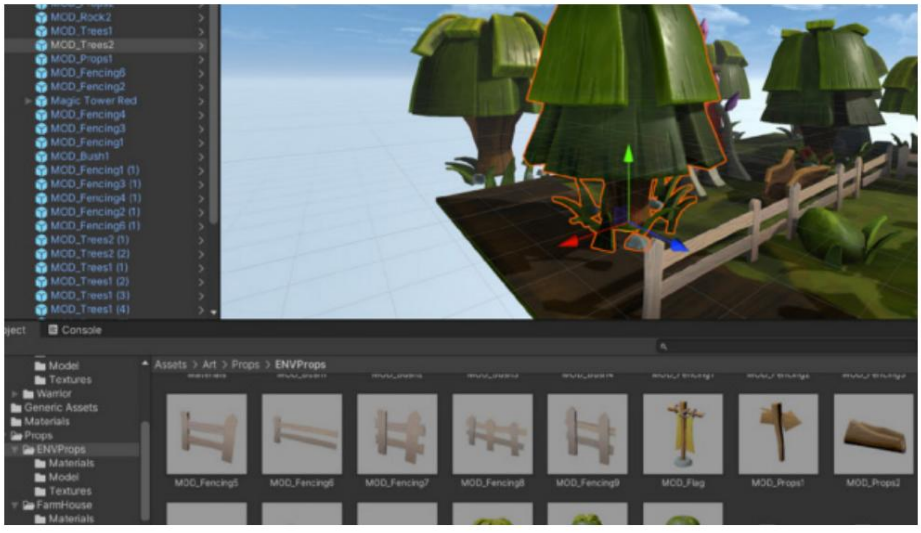


Un ejemplo de Prefabs, los bloques de construcción para un entorno

El sistema Prefab de Unity le permite crear, configurar y almacenar un GameObject, con todos sus componentes, propiedades y GameObjects secundarios, como un activo reutilizable.

El recurso prefabricado funciona como una plantilla a partir de la cual se pueden crear nuevas instancias prefabricadas en la escena. Estos recursos se pueden compartir entre escenas o otros proyectos sin necesidad de configurarlos nuevamente.

Los prefabs son editables. Puedes editar un prefab por objeto, modificando una sola instancia en la escena o aplicando los cambios a todas. Esto facilita la corrección de errores de objetos, la sustitución de ilustraciones o la realización de otros cambios.



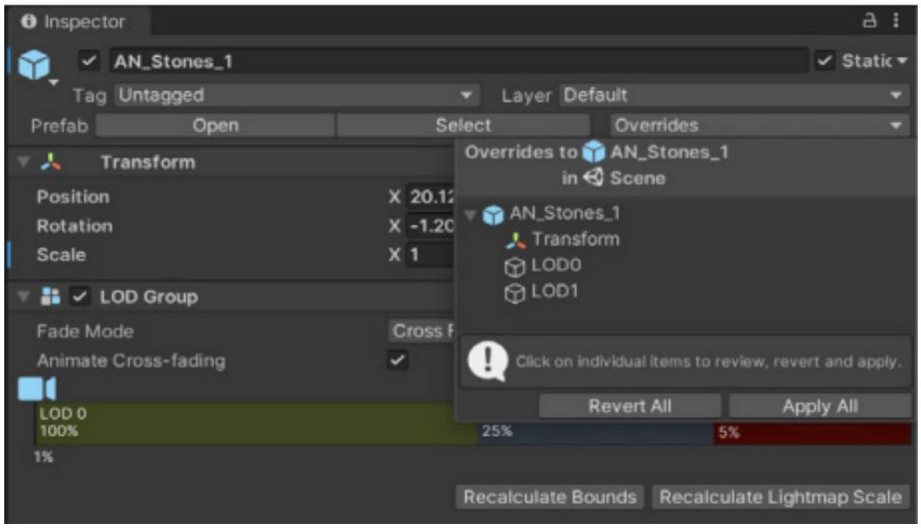
El Prefab seleccionado en la vista Jerarquía también se puede encontrar en la vista Proyecto como un activo, que puedes reutilizar tantas veces como sea necesario.

Las instancias modificadas de un prefabricado tienen una línea azul junto a las propiedades que anulan las del prefabricado original. Todas las anulaciones también se pueden mostrar simultáneamente mediante un menú desplegable. Las anulaciones se pueden transferir al prefabricado original.

© 2023 Unity Technologies

Machine Translated by Google

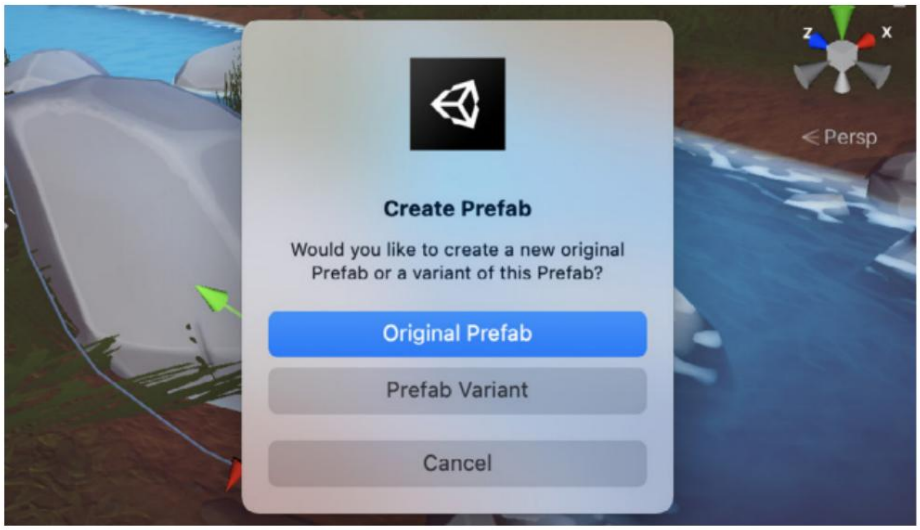
Activo mediante el comando Aplicar, que también transferirá los cambios a todas las demás instancias de Prefab. Otra opción es usar Revertir para restaurar los valores originales de Prefab, o simplemente dejar la instancia modificada en la escena.



El menú desplegable Anulaciones relacionado con los cambios de Escala (con una línea azul al lado); desde este menú, puede aplicar el cambio al Prefab o revertir los valores al Prefab original.

Los prefabs anidados permiten insertarlos entre sí para crear uno más grande. Por ejemplo, podría ser un edificio compuesto por prefabs más pequeños, como los de las habitaciones y el mobiliario. Esto facilita dividir el desarrollo de recursos entre un equipo de varios artistas y desarrolladores que pueden trabajar simultáneamente en diferentes partes del contenido.

Una variante de prefabricado permite derivar un prefabricado de otros. Las variantes de prefabricado son útiles cuando se desea tener un conjunto de variaciones predefinidas de un prefabricado, por ejemplo, para crear variantes de un personaje enemigo con diferentes estadísticas o materiales. Para crear la variante, arrastra un prefabricado existente modificado en la escena a la vista del proyecto.



El menú emergente cuando arrastra un Prefab modificado a la vista del Proyecto

© 2023 Unity Technologies

Machine Translated by Google

Una variante prefabricada hereda las propiedades de otro prefabricado, llamado base.
Las anulaciones realizadas en la variante de prefabricado tienen prioridad sobre las del prefabricado base.
valores. Una variante de prefabricado puede tener cualquier otro prefabricado como base, incluido el modelo
Prefabricados u otras variantes de prefabricados. También puede eliminar todas las modificaciones y
volver al Prefab base en cualquier momento.

Los prefabricados anidados y las variantes también funcionan bien con los sistemas de control de versiones. Equipo
Los miembros pueden trabajar simultáneamente en diferentes Prefabs y actualizar sin conflictos.
y permitir a los desarrolladores mantener siempre una copia de seguridad de las diferentes partes.

3D o 2D

Puedes crear juegos tanto en 3D como en 2D en Unity o incluso combinar ambos gráficos
estilos en el mismo proyecto. El Editor, los flujos de trabajo de codificación y algunas herramientas funcionan en conjunto.
De la misma manera en ambas perspectivas. Sin embargo, existen diferencias y
son herramientas específicas para cada tipo de juego, como Terrain para 3D o Tilemap para 2D.
Los recursos 2D y 3D también se gestionan de forma diferente en Unity. La siguiente tabla enumera
Algunas diferencias y similitudes entre las dos perspectivas.

Área de características	Proyecto 3D	Proyecto 2D
Bienes comunes	Modelo 3D	Duende
Canalizaciones de renderizado	Canal de renderizado integrado URP, HDRP	Canal de renderizado integrado Urgencia
Prototipado	ProBuilder	Editor de mapas de mosaicos 2D
Iluminación	Luces	Luces 2D
Física	Física	Física 2D
manipulación de personajes	Modelo premontado en DCC o aparejo de animación	Animación 2D
Diseño de entornos	Terreno	Editor de mapas de mosaicos 2D
Estrías	Estrías	Forma de sprite 2D
Animación	Ventana de animación, Controlador de animación	
Sistemas de cámaras	Cinemachine	
Efectos especiales	Sistema de partículas y gráfico de efectos visuales	
Creación de sombreadores	Gráfico de sombreado	
Codificación	Scripting visual en C# y Unity	
[illegible]	Interfaz de usuario de Unity y kit de herramientas de interfaz de usuario	
Controles de entrada	Entrada heredada o sistema de entrada	

Activos 3D

Una malla 3D es la construcción estructural de un modelo 3D compuesto de múltiples polígonos
formas. En Unity, se deben agregar dos componentes a un GameObject para
adjuntarle un modelo 3D y luego renderizarlo en la pantalla: un filtro de malla y un
Renderizador de malla.

Machine Translated by Google

Los materiales definen cómo renderizar la superficie de la malla 3D, combinando información sobre la apariencia visual de la superficie, como texturas, tonos de color y sombreadores.

Sombreadores son una serie de instrucciones que se ejecutan en la GPU que determinan cómo Unity muestra GameObjects en pantalla. Cada canal de renderizado En Unity viene con sombreadores prediseñados, por lo que cada píxel se renderiza en función de la entrada de iluminación y la configuración del material.

Formatos de archivo

Unity utiliza el formato de archivo FBX internamente, por lo que se recomienda que también lo utilice siempre que sea posible en su producción para evitar formatos de archivos de modelos propietarios. Por ejemplo, si usa Blender y guarda sus archivos .blend en la carpeta Asset del proyecto Unity, todos los demás que trabajen en el proyecto tendrán que instalar y usar la misma versión de Blender.

Unity importa mallas Desde otro software DCC, con todos los nodos en su posición, rotación y escala guardadas. También se importan los puntos de pivote y sus nombres, junto con los vértices, polígonos, triángulos, UV, normales, huesos, mallas con revestimiento y animaciones. Al iterar sobre recursos en otro software DCC compatible, Unity actualizará el GameObject correspondiente y reflejará los cambios en el Editor de Unity cada vez que guarde el archivo.

Un GameObject de un modelo 3D a la izquierda y un sprite 2D a la derecha

Activos 2D

Los sprites son objetos gráficos 2D. Si estás acostumbrado a trabajar en 3D, los sprites son básicamente texturas, pero existen técnicas especiales para combinarlas y gestionarlas eficientemente durante el desarrollo. Para asegurarte de que Unity gestione las texturas como sprites al importarlas a tu proyecto, consulta Configuración del proyecto > Editor > Comportamiento predeterminado y selecciona 2D. Si iniciaste el proyecto desde una plantilla 2D, esta opción ya está configurada.

© 2023 Unity Technologies

Machine Translated by Google

Unity proporciona herramientas como el Editor de Sprites para configurar el activo de origen, lo que incluye configurar la resolución del activo, su malla o contorno, opciones de corte, diseño y aparejo, y agregar mapas normales y de máscara para iluminación 2D.

Para los recursos 3D, la resolución se define por el número de polígonos o el tamaño de la textura, mientras que para los recursos 2D se trabaja en PPU (píxeles por unidad), que indica a Unity la resolución del sprite por unidad. Si la cámara se acerca demasiado a la unidad, pero no hay suficientes píxeles de resolución, se observará pixelación; por otro lado, usar una resolución superior a la necesaria puede resultar en un desperdicio de rendimiento y memoria. Lee este blog. para obtener más información sobre la resolución de activos 2D.

Formatos de archivo

Se recomienda importar los sprites en formatos sin pérdida, como PNG. Unity admite los tipos de archivo de imagen más comunes, como BMP, TIF, TGA, JPG y PSD. Si guarda sus archivos de Photoshop con capas (.psd) en la carpeta "Recursos", Unity los importará como imágenes aplanadas a menos que instale el Importador PSD 2D. Paquete que permite leer la información de las capas, lo que hace que el proceso de importación sea más eficiente. Obtenga más información sobre el importador PSD 2D en este blog.

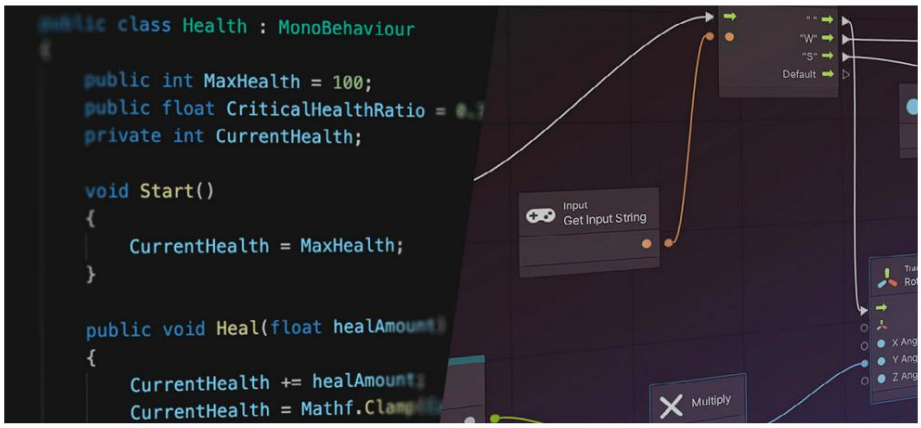
Dragon Crashers, una demostración 2D de Unity, está disponible en Unity Asset Store.

© 2023 Unity Technologies

Machine Translated by Google

Codificación

Hay dos maneras de crear la lógica de un juego en Unity: escribir scripts en C# o conectar y agrupar nodos y gráficos en el sistema de scripting visual de Unity. Es probable que los programadores proporcionen la mayor parte del código de un juego, pero como parte del proceso inicial de prototipado, puede ser útil para los diseñadores tener conocimientos básicos de cómo funcionan los scripting en Unity.

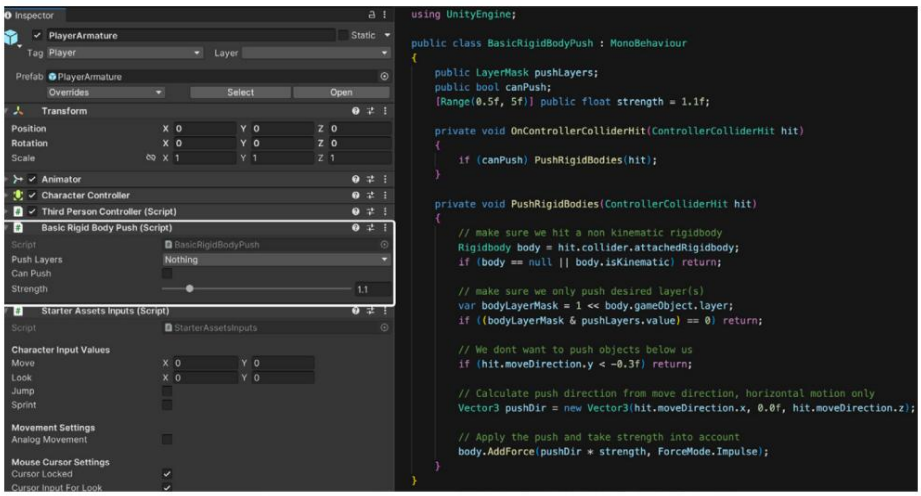


Un script de Unity C# a la izquierda y un script visual a la derecha

Descripción rápida de C#

Los scripts indican a los GameObjects cómo comportarse. Son los scripts y componentes asociados a los GameObjects, y cómo interactúan entre sí, lo que crea la jugabilidad.

Un script se conecta con el funcionamiento interno de Unity implementando una clase derivada de la clase integrada MonoBehaviour. Esta clase es utilizada por GameObjects y proporciona funciones que permiten desarrollar un juego. Unity no incluye una herramienta para escribir código ni IDE, pero existen paquetes en el Administrador de Paquetes que permiten usar Microsoft Visual Studio . Código, o JetBrains Rider.



Un componente de Unity resaltado en el Inspector (izquierda) y cómo se ve en el código (derecha)

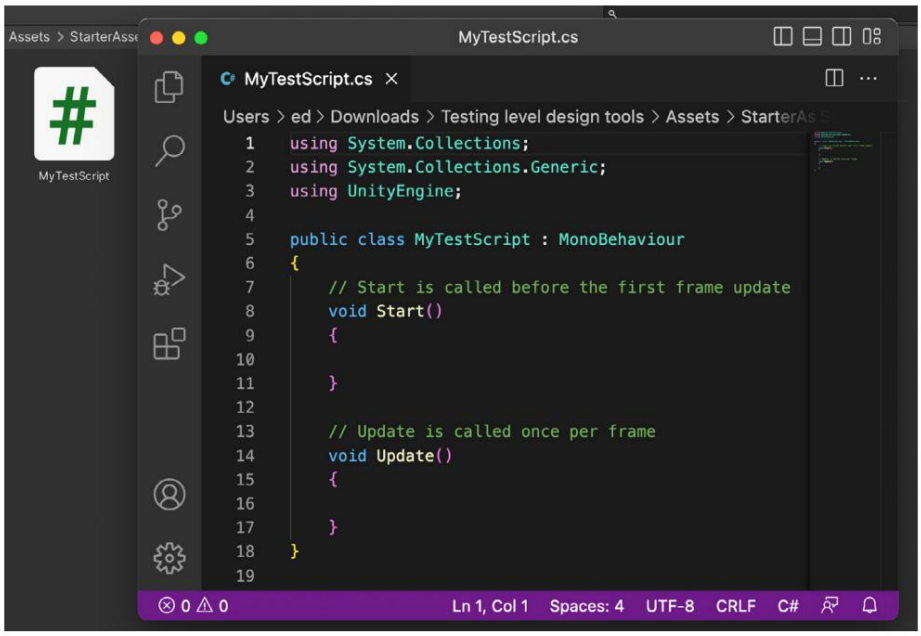
© 2023 Unity Technologies

Machine Translated by Google

Creando un nuevo script

Puede crear un nuevo activo de script en el Editor a través de Activos > Crear > Script de C# o en la ventana del Inspector, a través de Agregar componente > Nuevo script.

Cuando Unity crea un nuevo activo de script, rellena previamente el script con algún código predeterminado.



Un recurso de script de Unity en el IDE Microsoft Visual Studio Code

Echemos un vistazo con más detalle al recurso de script anterior:

La palabra clave using importa un espacio de nombres. Los espacios de nombres incluyen clases y funciones que puedes usar en tu código. Es necesario usar UnityEngine para usar las funciones de Unity. como Inicio o Actualizar.

Los comentarios tienen dos barras diagonales al principio de la línea, y es habitual explicar brevemente la funcionalidad que se intenta lograr en el código antes de las variables o funciones. Unity siempre ignora los comentarios al ejecutar el juego.

— El nombre de la clase (que debe coincidir con el nombre del archivo del activo) se deriva de Clase MonoBehaviour. Permite adjuntar el script al GameObject como componente. Todo dentro de esta clase está entre corchetes, variables y funciones.

Hay algunas funciones de evento especiales que Unity ejecuta automáticamente al ejecutar el juego. La función Start se ejecuta automáticamente al cargar el GameObject, y la función Update se ejecuta cada vez que se renderiza un fotograma del juego. Puedes ver el orden de ejecución de las funciones de evento en

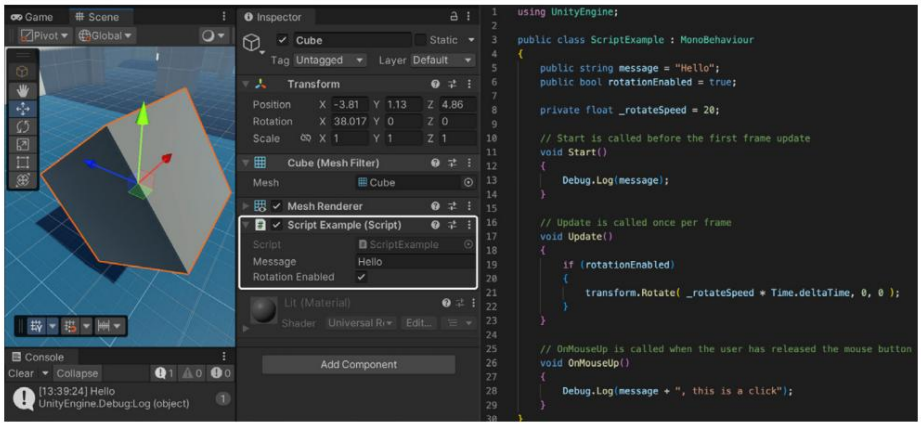
Este gráfico detallado. Deberás escribir tus funciones en el lugar correcto según tus necesidades. Por ejemplo, es habitual almacenar referencias a otros componentes como variables solo una vez, dentro de la función de inicio, para usarlas posteriormente en la función de actualización.

© 2023 Unity Technologies

Machine Translated by Google

Un ejemplo de guión

En el ejemplo a continuación, el objeto cubo tiene un script adjunto (resaltado en la ventana del Inspector en la imagen) que gira el cubo e imprime un mensaje en la consola. Analicemos este script con más detalle:



Un script simple en el Editor y en el IDE

La clase ScriptExample contiene variables, como una cadena, un valor booleano y un valor de punto flotante. Estas son variables públicas , lo que significa que se mostrarán en el Inspector para su modificación y acceso desde otros scripts. Las variables privadas no se muestran en el Inspector y otros scripts no pueden acceder a ellas. Tanto las variables públicas como las privadas tienen un valor predefinido en la declaración.

— Dentro de la función Inicio , que se invoca solo una vez, hay una instrucción para imprimir un mensaje en la ventana de la consola.

La función Actualizar contiene lógica básica con una sentencia if . Si la condición es verdadera, el cubo rotará en el eje X a _rotateSpeed multiplicado por Time.deltaTime para que la velocidad de fotogramas del movimiento sea independiente. Transform es una forma abreviada de acceder a las propiedades Transform del GameObject con el script adjunto. Esto modifica la rotación del objeto y se ejecuta en cada fotograma.

— Se ha añadido una función de Evento (que no viene con la función predeterminada) código), que activará Unity al hacer clic con el ratón. A continuación, el script solicita que se imprima otro mensaje en la ventana de la consola.

Idealmente, un script debería intentar resolver un solo objetivo. Este es el principio de responsabilidad única en programación: cada módulo, clase o función es responsable de una cosa y encapsula solo esa parte de la lógica.

Esto le permite mantener su código modular. Los scripts modulares con una sola responsabilidad son más fáciles de reutilizar, ampliar y probar en otros sistemas. Si su script intenta resolver más de una tarea, podría funcionar mejor si se divide en scripts más pequeños con funcionalidades independientes.

Puede leer más sobre la aplicación de principios de programación y patrones de diseño en proyectos de Unity en esta publicación de blog.

© 2023 Unity Technologies

Machine Translated by Google

Mejora tu código con patrones de programación de juegos

Este libro electrónico explica patrones de diseño conocidos y comparte ejemplos prácticos para usarlos en su proyecto de Unity.

Al implementar patrones de diseño de programación de juegos comunes en su proyecto de Unity, puede crear y mantener de manera eficiente una base de código limpia, organizada y legible, lo que a su vez crea una base sólida para escalar su juego, su equipo de desarrollo y su negocio.

Descargar el libro electrónico

Recursos de aprendizaje

Programar en Unity es un tema muy amplio. Cualquier funcionalidad de juego se puede crear con scripts, al igual que herramientas de edición personalizadas y, mediante las API de Unity, cualquier objeto en tiempo de ejecución. Si eres completamente nuevo en programación de scripts, puedes empezar con algunos recursos para principiantes:

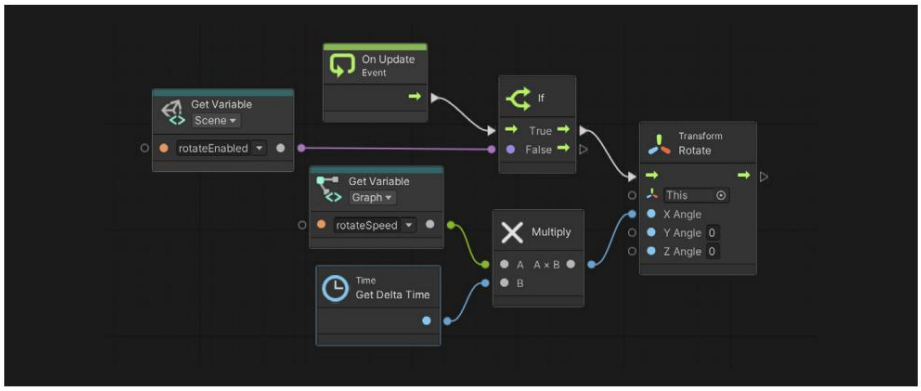
— Programación en C# en Unity para principiantes

— Tutoriales de scripting para principiantes de Unity Learn

— Creación y uso de scripts del manual de Unity

Scripting visual de Unity

Scripting visual de Unity es una herramienta gráfica basada en nodos que programadores y no programadores pueden usar para diseñar la lógica y la interactividad del juego, tanto en la creación de prototipos como en la producción de juegos, sin escribir código.



Durante el juego, puedes ver el flujo de datos visualmente en Unity Visual Scripting. Los nodos presentan una flecha verde que representa el flujo de ejecución de izquierda a derecha, el puerto naranja representa la variable y los puertos grises representan el valor de entrada y salida de la variable.

Machine Translated by Google

Visual Scripting tiene algunas características clave que resultan útiles para los diseñadores:

— Nodos visuales: Crear lógica con nodos le permite centrarse en la configuración
establecer conexiones válidas, lo que ayuda a reducir potencialmente el ensayo y error.
Además, puedes ver qué nodos no están activos y el flujo lógico.
entre los nodos y los datos transmitidos. Aprenda sobre los nodos. en los documentos.

— Lógica simplificada: un gráfico de estados Es un sistema de máquina de estados. Permite a los usuarios
Para definir visualmente qué lógica ejecutar según el estado activo del objeto
y cuándo pasar a otro estado. Un gráfico de estados es útil para los objetos.
que requieren ramificaciones lógicas como la IA de un enemigo.

— Cambios en los nodos en el modo de reproducción: cambia los comportamientos sobre la marcha mientras
El juego se está ejecutando en lugar de tener que salir primero del modo de juego.

Creación de un nuevo gráfico de script

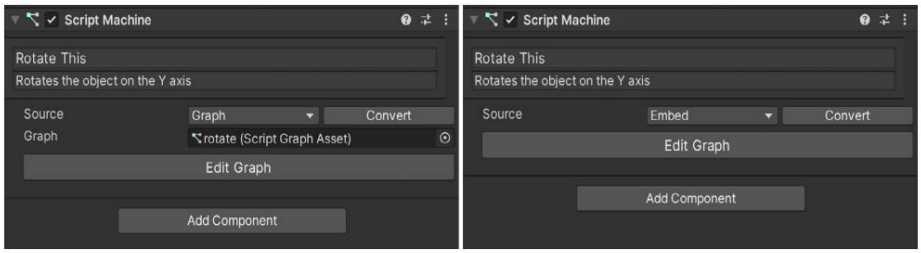
Gráficos son representaciones visuales de la lógica y, por lo tanto, son el núcleo de la lógica visual.
Scripting. Hay dos tipos de gráficos:

— Los gráficos de secuencia de comandos (flujo) conectan acciones y valores individuales en un contexto específico.
orden. El orden de ejecución es el flujo del script.

— Los gráficos de estados crean diferentes estados y las transiciones entre ellos.

Para crear su primer gráfico de script (comportamiento) para su GameObject, vaya a Ventana
> General > Jerarquía, o presione Ctrl+4 (macOS: Cmd+4) para abrir la Jerarquía
ventana. En la jerarquía, seleccione el GameObject, haga clic en Agregar componente y
Seleccione Script Machine.

Desde aquí, puede indicar si desea reutilizar un comportamiento existente, seleccionar un
Utilice el activo de gráfico de script como fuente o comience con un nuevo comportamiento con el
fuente incrustada. Siempre puedes convertir el comportamiento incrustado en un
activo con el botón Convertir .



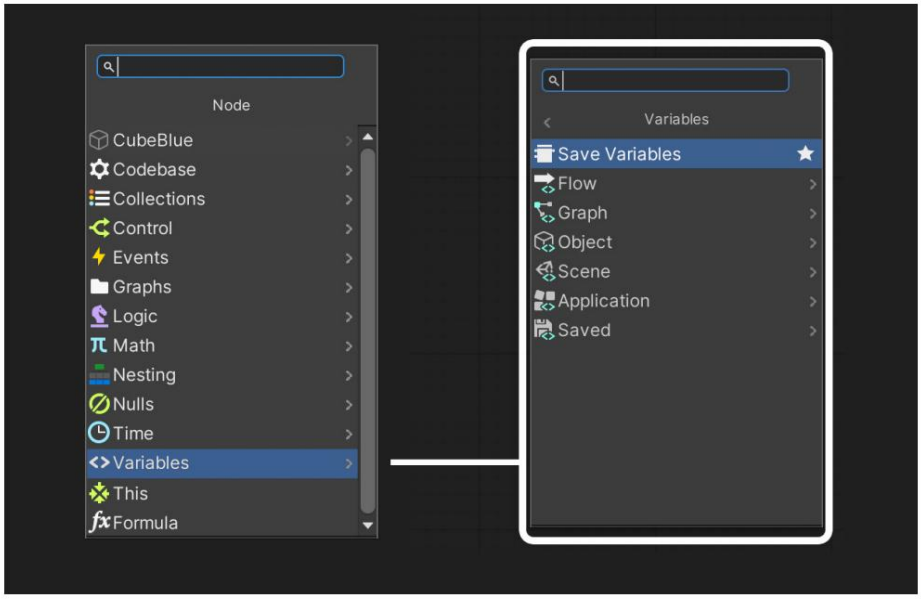
El componente que utiliza un activo de Script Graph a la izquierda y un comportamiento autónomo a la derecha

Para crear su primer comportamiento, abra el gráfico con Editar gráfico o abriendo el
Activo en la vista del Proyecto. El Editor de Gráficos Estará vacío, por lo que deberá agregarlo
Algunos nodos para crear la lógica. Probablemente querrás agregar primero un nodo de evento.
ya que esto desencadenará el comportamiento.

Encontrará una lista completa de todos los nodos disponibles en la referencia de Nodos.
sección en la documentación de Visual Scripting.

© 2023 Unity Technologies

Machine Translated by Google

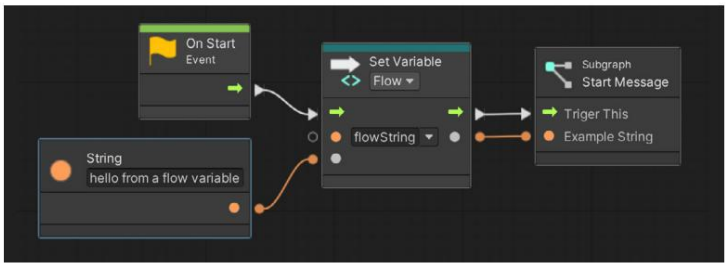
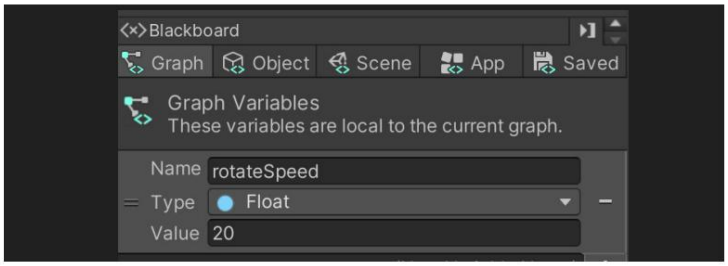


Creando un nuevo nodo con las opciones para Nodos variables a la derecha

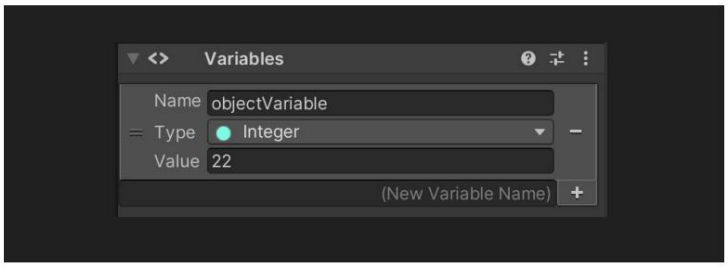
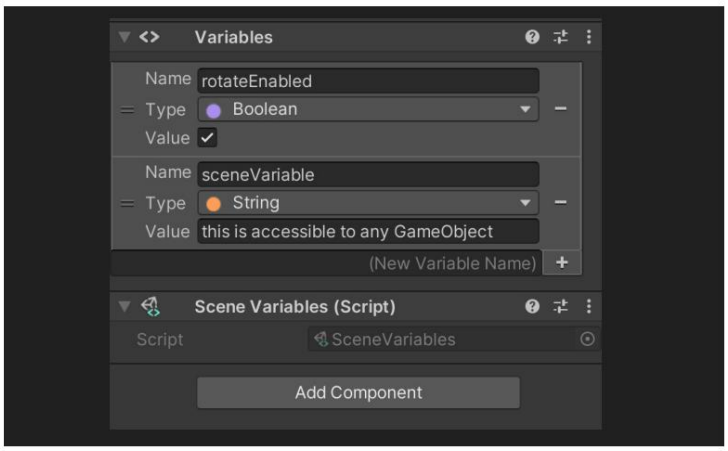
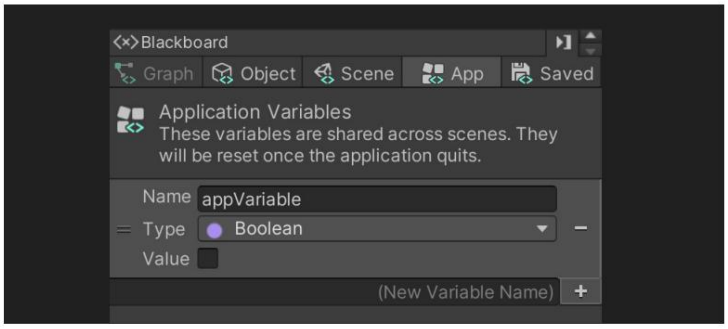
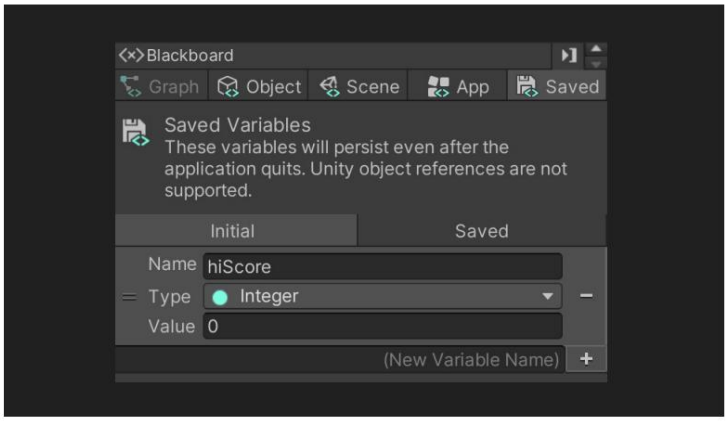
Variables en Visual Scripting

Para crear lógica, es necesario trabajar con variables, que actúan como contenedor de información que puede cambiar durante la ejecución de una aplicación. Una variable necesita un nombre, el tipo de datos que contiene y su valor predeterminado. Las variables también tienen ámbitos. El ámbito de una variable determina qué partes de su Script Graph pueden acceder a qué variables para leer o modificar sus valores. El ámbito también puede decidir si otro Script Graph puede acceder a una variable.

La siguiente tabla muestra los diferentes alcances de las variables.

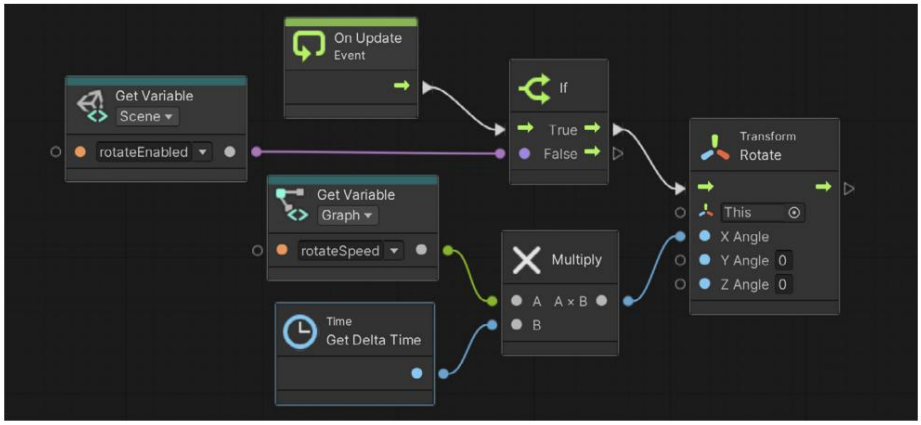
Alcance	Descripción	Visualización en Unity
Variable de flujo	Necesita ser creado y utilizado directa o indirectamente en el mismo flujo; las variables de flujo tienen el alcance más pequeño	 Sólo accesible desde el Editor de gráficos
Variable gráfica	Disponible para cualquier nodo del mismo Gráfico de guiones; puedes crearlos y agregarlos desde Blackboard elemento	 Accesible desde el área de Blackboard

Machine Translated by Google

La variable de objeto puede ser utilizada por cualquier gráfico de script. Subgráfico y gráfico de estados adjuntos a un GameObject específico		 Accesible desde el área de Blackboard y el Inspector de GameObject
Variable de escena	Pertenece a la escena actual; accede a las variables de la escena desde cualquier Script Gráfico adjunto a un diferente GameObject en una sola escena	 Una variable de escena
Variable de la aplicación	Pertenece a una aplicación completa, accesible a través de múltiples escenas mientras se ejecuta la aplicación	 Accesible desde el área de Blackboard
Variable guardada	El mismo alcance que una variable de aplicación, pero con la diferencia de que persistir si la aplicación se cierra; llamado Preferencias del jugador en la unidad	 Accesible desde el área de Blackboard

© 2023 Unity Technologies

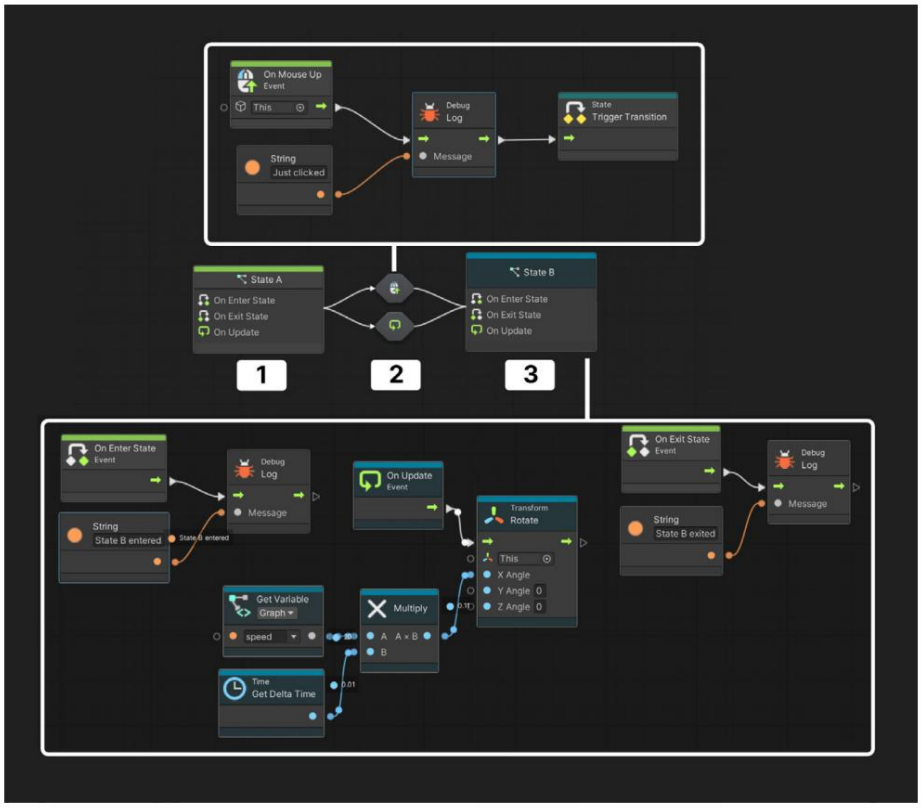
Machine Translated by Google



Una variable de escena y una variable de gráfico utilizadas en un gráfico de script simple que rota el GameObject adjunto

Máquinas de estados con gráfico de estados

Los gráficos de estado son componentes adjuntos al GameObject, cuya lógica puede ser un recurso o estar integrada en él. Se crean de la misma manera que un gráfico de script. Los gráficos de estado crean comportamientos de la IA o la estructura de la escena. Los estados indican al objeto cómo debe comportarse en ese estado. Por ejemplo, piense en comportamientos como "patrullar" o "perseguir" a una IA enemiga, o en una puerta que está en los estados "cerrada", "desbloqueada" o "abierta". La lógica para pasar de un estado a otro reside en las transiciones. En esencia, gráficos de script que pueden activar la transición. Por ejemplo, cuando el jugador se acerca demasiado al enemigo, el estado cambia de patrulla a persecución.



© 2023 Unity Technologies

Machine Translated by Google

Repasemos el gráfico de estados en la imagen de arriba:

1. Este es el estado inicial, marcado con la barra verde en la parte superior del nodo (puede activar un estado como inicial haciendo clic derecho sobre él). Un nuevo estado crea tres eventos: Entrar, Salir y Actualizar. Puede usarlos para empezar a crear la lógica.

2. Se trata de dos transiciones. Una activa el estado de transición al hacer clic en el objeto (mostrado arriba), y la otra evalúa una condición específica en Actualizar y realiza la transición cuando se cumple. El icono en la forma hexagonal del gráfico de transición ofrece una pista visual del evento que utilizará el comportamiento de la transición.

3. Este es el estado activo actual, indicado por la línea de encabezado azul. Se imprimen mensajes de depuración al entrar y salir de este estado. En este estado, el objeto está rotado (como se muestra a continuación).

El scripting visual y los scripts de C# también pueden funcionar juntos en el mismo proyecto. Por ejemplo, los programadores del equipo pueden crear nodos de scripting visual personalizados para que los diseñadores trabajen en la funcionalidad en un entorno controlado. Por ejemplo, muchos estudios suelen pedir a sus desarrolladores que creen nodos de scripting visual personalizados para que los diseñadores puedan usar fácilmente la funcionalidad o los eventos gestionados por el código base del juego.

Esta presentación Muestra las posibilidades de Unity Visual Scripting para programadores.

Más recursos

— Página de scripts visuales de Unity

— Aplicación de Visual Scripting: Clive the Cat's

— Documentación de Visual Scripting

— Unity Visual Scripting para artistas y diseñadores

Física

Unity proporciona un conjunto completo de sistemas de física 3D y 2D para una simulación física realista. Física 3D integrada. El sistema se utiliza para GameObjects basados en malla. La física 2D El sistema utiliza GameObjects basados en sprites.

Equilibrar y ajustar los parámetros físicos es fundamental en el diseño de una jugabilidad. Casi todos los juegos requieren simulaciones e interacciones físicas, como la gravedad, las colisiones de objetos, las reacciones entre objetos, etc.

© 2023 Unity Technologies

Machine Translated by Google

Una imagen de Creator Kit: Puzzle, Un proyecto tutorial para nuevos usuarios de Unity que está disponible en Unity Asset Store

Creando colisiones

Componente del colisionador

Colisionadores Se aplican a GameObjects para representar su forma física en una simulación de física. Existen componentes de colisión de forma primitiva, como Caja, Cápsula, Esfera o Rueda, que se pueden modificar con el botón "Editar colisionador" . Estas formas primitivas suelen ser suficientes para iterar sobre las mecánicas básicas del juego.

El sistema de física utiliza la unidad Unity como referencia para replicar la física del mundo real. La escala de la unidad equivale a un metro. Los objetos de diferentes tamaños deben modelarse a una escala precisa. Un personaje humano, por ejemplo, debe tener una altura aproximada de dos unidades. Es importante usar el tamaño de malla adecuado para tu GameObject: un rascacielos en ruinas se desmoronará de forma diferente en una escena que una torre hecha de bloques de juguete.

El componente Collider define los límites físicos del objeto. Al añadir un Static Collider a un GameObject, el sistema de física trata el objeto como sólido e inamovible.

Componente de cuerpo rígido

Para simular comportamientos físicos como el movimiento, la gravedad, las colisiones y las articulaciones, es necesario configurar los elementos de la escena como cuerpos rígidos. Esto se logra añadiendo un Rigidbody. Componente para un GameObject. El componente Rigidbody proporciona un método basado en la física para controlar el movimiento y la posición de un GameObject.

Los GameObjects con un componente Rigidbody pueden colisionar de forma realista con objetos estáticos. Los objetos Rigidbody dinámicos también pueden colisionar entre sí, por ejemplo, como dos bolas de billar chocando.

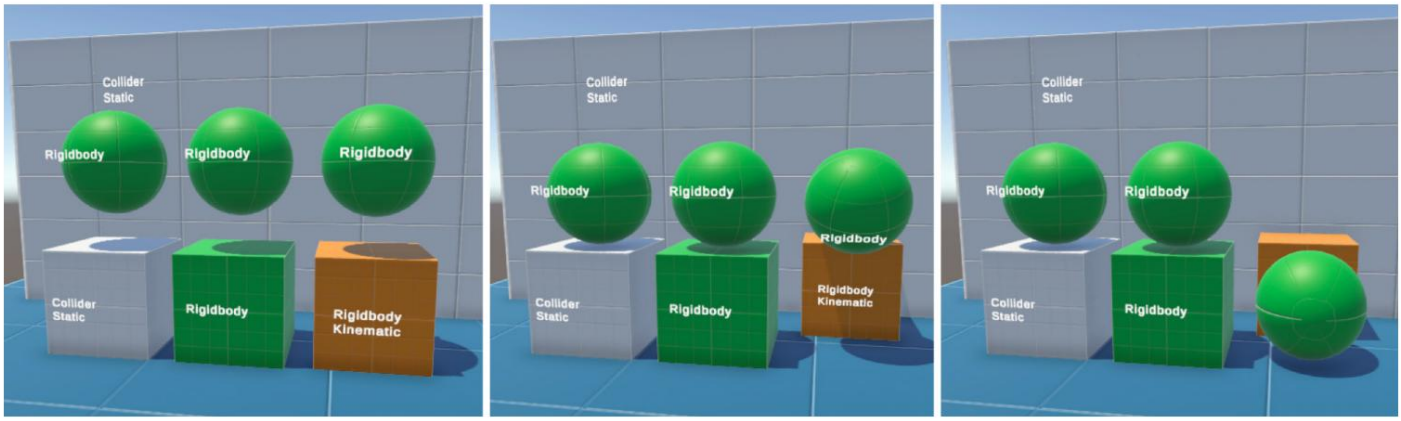
© 2023 Unity Technologies

Machine Translated by Google

Veamos dos configuraciones importantes en el componente Rigidbody:

La propiedad "Es Cinemático" permite que el Rigidbody afecte a otros objetos mediante la física, pero no se verá afectado. Por ejemplo, un avatar de mano en un juego de RV puede interactuar con objetos mediante la física, pero no se desea que la física actúe sobre la mano ni sobre una plataforma móvil a la que el jugador deba saltar.

La propiedad " Usar Gravedad" es, como su nombre indica, la fuerza de gravedad que afecta al GameObject. Si no se activa, el objeto puede ser empujado por otros, pero parecerá ingrávido, ya que no hay desaceleración debido a la fuerza de gravedad.



Las esferas verdes de la imagen superior son cuerpos rígidos dinámicos capaces de interactuar con otros objetos dinámicos, estáticos o cinemáticos. En el modo de juego, se puede observar cómo la gravedad simulada atrae a los cuerpos rígidos dinámicos hacia abajo, de modo que colisionan con los demás elementos. El cubo naranja es un objeto cinemático; un script lo mueve hacia adelante y hacia atrás sin reaccionar a otras fuerzas, lo que hace que atraviese la pared estática y no oponga resistencia al empujar la esfera verde.

A medida que se acerque al funcionamiento de la colisión física en Unity, encontrará cierta terminología común en los tutoriales y la documentación.

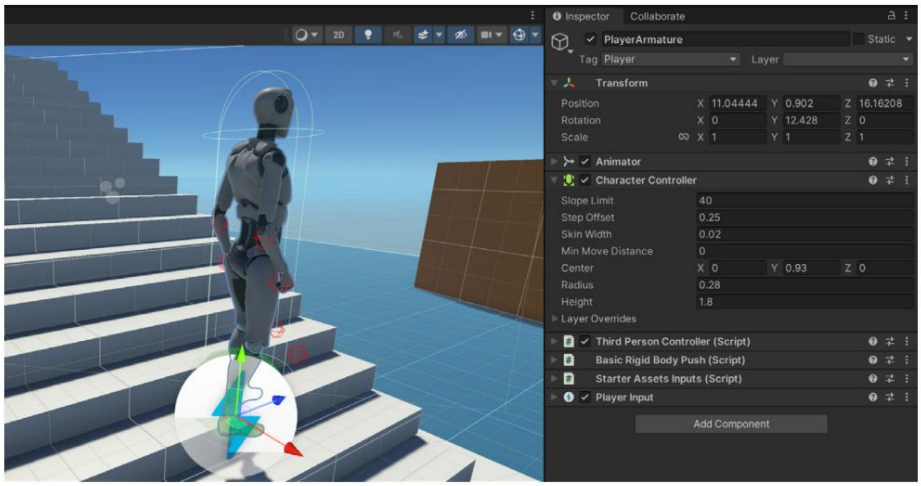
Componentes de terminología necesarios	Movimiento Común implementación	Caso de uso
Estático colisionador	Colisionador, es decir, colisionador de cajas, colisionador de esferas, etc.	Ninguno	Agregado durante el proceso de diseño de niveles en el Vista de escena	Muros, límites, grandes apoyos, terreno, objetos estáticos
Cinemático Cuerpo rígido Colisionador	Colisionador y Cuerpo rígido (es Cinemático propiedad habilitada)	Sí, pero no reaccionar a externo efectivo	Movimiento patrón implementado en el código o con animación	Plataformas móviles, grandes accesorios animados, personajes o objetos que están programados o en un juego de realidad virtual, por ejemplo, una mano virtual que puede interactuar con objetos

© 2023 Unity Technologies

Machine Translated by Google

Cuerpo rígido Colisionador o dinámica Cuerpo rígido	Colisionador y Cuerpo rígido (es Cinemático propiedad discapacitada)	La masa, la gravedad y la resistencia afectarán La forma en que reacciona a fuerzas; choca con estática, cinemática y objetos dinámicos	Configurar en el Vista de escena para inanimado objetos o aplicar fuerzas o velocidad hacia ellos en código	Elementos de juego basados en la física como objetos de rompecabezas interactivos objetos, o accesorios que necesitan reaccionar a la gravedad
Personaje Controlador	Personaje Controlador	Preciso movimiento (aunque no siempre realista); reacciona a efectivo, choca con estática, cinemática, y objetos dinámicos	Controlado por la entrada del jugador en código	Interpretable personajes donde la precisión Se necesita más simulación de física realista, es decir, En una primera tirador personal juego

En la mayoría de los casos, un componente llamado Controlador de personajes Se usará para el personaje del jugador. El controlador de personaje permite que un personaje 3D interactúe con el mundo físico que lo rodea sin comportarse necesariamente de forma realista (dependiendo del estilo o género del juego, a menudo querrás darle al jugador más control y precisión de lo que es realista).



Proyectos de muestra de Unity Controlador de personajes en tercera persona y controlador de personajes en primera persona Incluye un controlador de personaje jugable ya configurado.

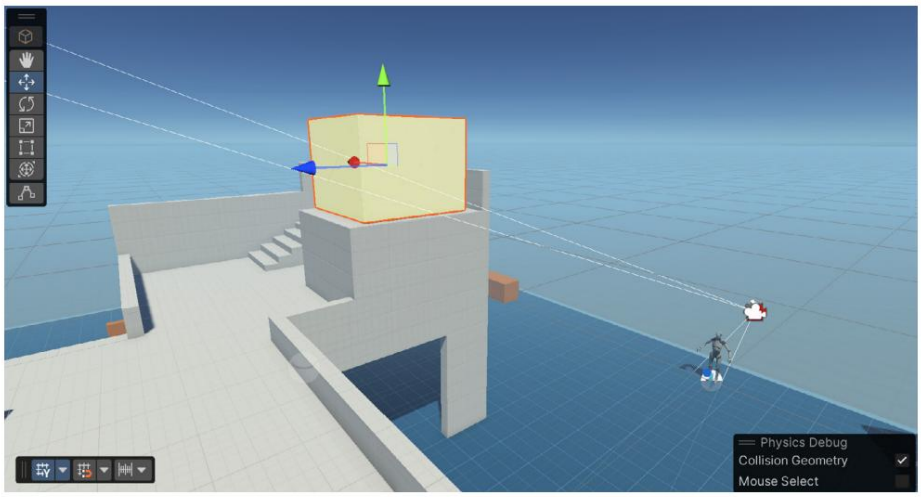
© 2023 Unity Technologies

Machine Translated by Google

Colisionadores de gatillo

Un colisionador configurado como disparador (mediante la propiedad "Es disparador") se puede usar cuando un objeto necesita detectar o reaccionar a otro sin contacto físico. Esta propiedad permite que el/los objeto(s) en cuestión actúen como un sensor. Algunos ejemplos de dónde se usaría la propiedad "Es disparador" son para designar áreas de entrada, guardado automático o salida.

La implementación de la funcionalidad en el código es muy similar al escenario de colisión con cuerpos rígidos, pero esta vez los objetos recibirán mensajes de activación (OnTrigger) en lugar de mensajes de colisión (OnCollision) en caso de colisión.



Los colisionadores de activación pueden reconocer cuándo un objeto entra, se queda o sale del colisionador. En este ejemplo, una escena o secuencia al final de un nivel se activa cuando el jugador llega a la cima del edificio al entrar en el cubo amarillo.

Mensajes de activación se producirá cuando un colisionador estático, de cuerpo rígido o cinemático con la opción Is Trigger habilitada interactúa con un colisionador de cuerpo rígido y cinemático sin disparador. Los colisionadores con disparador estático no interactuarán entre sí ni con un colisionador estático sin disparador.

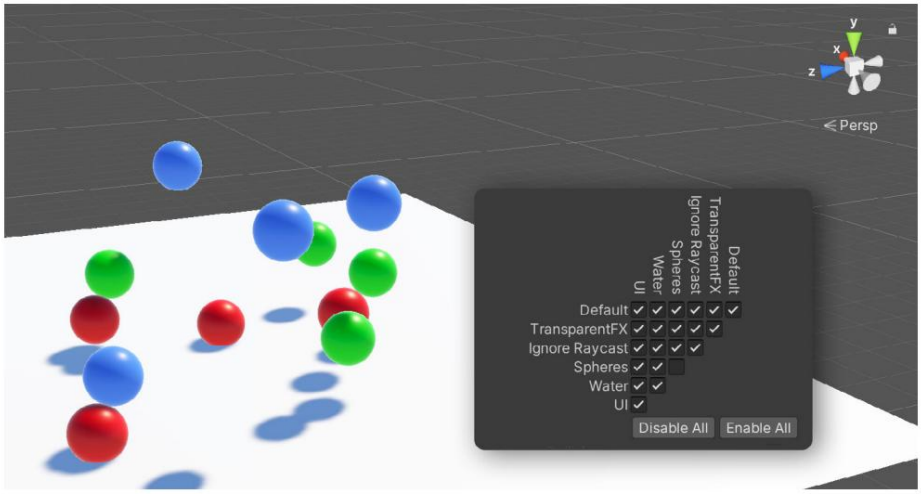
Capas de física

Colisiones basadas en capas Proporciona una manera de hacer que un GameObject colisione con otro GameObject configurado en una o más capas específicas. En la Configuración del Proyecto, encontrarás una Matriz de Colisiones que define qué GameObjects colisionan con qué capas. Por ejemplo, puedes definir que los objetos bala de la capa "Balas" no colisionen entre sí en la misma capa, pero sí con los objetos de la capa "Enemigos Terrestres".

Consejo: para mantener el rendimiento físico, habilite solo las capas de colisión que necesite.

© 2023 Unity Technologies

Machine Translated by Google

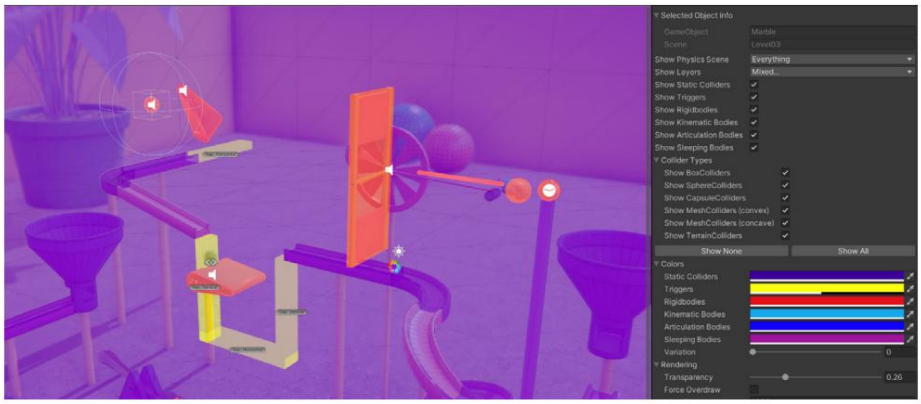


Las esferas de colores en la capa Esferas chocan con el piso en la capa predeterminada, pero no consigo mismas.

Diseño de niveles con física

A medida que llenas escenas grandes con una mezcla de objetos estáticos y dinámicos, puede resultar difícil probar tu juego y comprender por qué ocurren errores, como un personaje que sale de los límites del nivel o un objeto que no actúa como disparador cuando debería.

Puede habilitar el depurador de física En el Editor, en Ventana > Análisis > Depurador de Física (asegúrese de tener visibles los Gizmos de Escena). El Depurador de Física mostrará y resaltará los diferentes tipos de colisionadores y componentes de cuerpo rígido en la escena. Puede optar por ocultar algunos tipos y filtrar por capas de Física. Esto hace que sea eficiente encontrar la causa raíz de un problema, incluso con muchos objetos en la escena.



Desde la ventana del modo de depuración de física, puedes personalizar la configuración visual y especificar los tipos de GameObjects que quieres ver u ocultar en el visualizador.

Los diseñadores de juegos utilizan habitualmente el depurador de física para detectar rápidamente los problemas de colisión. La depuración de física también es útil si se desarrolla con productos como UModeler. o la herramienta ProBuilder en Unity.
A medida que modifica las mallas, es posible que olvide actualizar la configuración de colisión; el depurador lo ayudará a identificar dónde se necesitan las actualizaciones.

© 2023 Unity Technologies

Machine Translated by Google

En Unity, los GameObjects pueden implementar código en sus scripts que se ejecutará en cada fotograma renderizado, conocido como ciclo de actualización . En un juego que se ejecuta a 60 fps, el código dentro del ciclo de actualización se ejecutará 60 veces por segundo. El problema del código relacionado con la física es que no siempre tarda el mismo tiempo en renderizar un fotograma. A veces, incluso se pueden omitir fotogramas si hay una ralentización, lo que lo hace poco fiable para el comportamiento determinista de la física. El ciclo FixedUpdate se utiliza para los cálculos de física. Por defecto, FixedUpdate se produce cada 0,02 segundos (50 llamadas por segundo). Dependiendo de las necesidades de tu juego, será fundamental colaborar estrechamente con los programadores de tu equipo para configurar los ajustes del proyecto adecuados.

También se puede lograr una lógica independiente del marco en el ciclo de actualización con el parámetro Time.deltaTime. Supongamos que se desea aplicar una progresión suave en el tiempo a un objeto que cambia su posición linealmente cada segundo. Multiplicar por Time.deltaTime garantizará que la lógica tenga en cuenta el tiempo de renderizado del último fotograma para evitar que se vea inestable durante el ciclo de actualización. Si el fotograma anterior tarda más en renderizarse, el valor de multiplicación de Time.deltaTime será mayor para compensarlo.

```
void Update()
{
    Vector3 position = transform.position;
    position.x += SpeedValue * Time.deltaTime;
    transform.position = position;
}
```

Un ejemplo de script de cómo se pueden usar FixedUpdate y Time.deltaTime para mover un objeto linealmente a lo largo del tiempo

Interpolaciones Ayudará a evitar el tirones en el juego que resultan de combinar eventos FixedUpdate y Update regulares debido a las diferentes frecuencias de actualización. Puedes aplicar las propiedades de interpolación y extrapolación en el Rigidbody del GameObject visible y en movimiento. Lee sobre esta técnica en este blog de la comunidad.

La física 3D en Unity ofrece muchas posibilidades para crear juegos. Descubre la física de articulaciones, muñecos de trapo y más en la documentación de física.

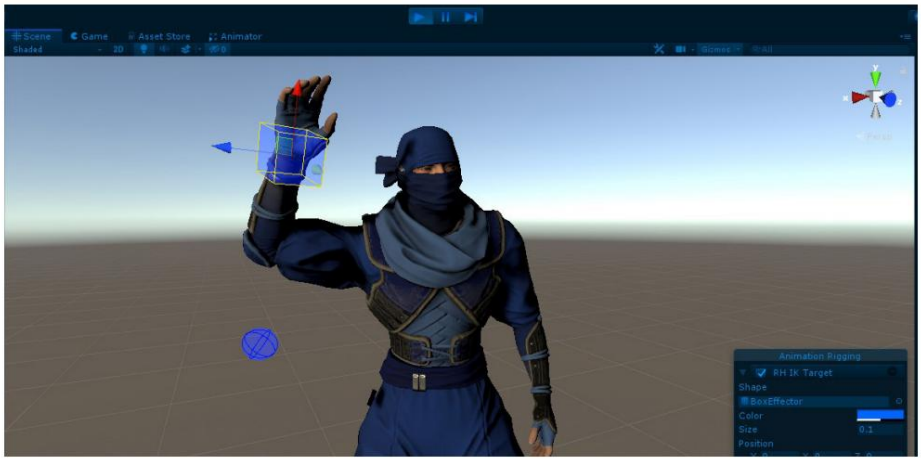
Existen componentes físicos específicos para proyectos 2D, como Collider 2D, Rigidbody 2D y Joints 2D. Los principios son los mismos que en 3D, y la implementación también es similar. Tenga en cuenta que los sistemas 3D y 2D no interactúan entre sí, por lo que deberá elegir una u otra perspectiva. Obtenga más información sobre física 2D en la documentación.

Obtenga una descripción general de las características de la física 2D en este video.

© 2023 Unity Technologies

Machine Translated by Google

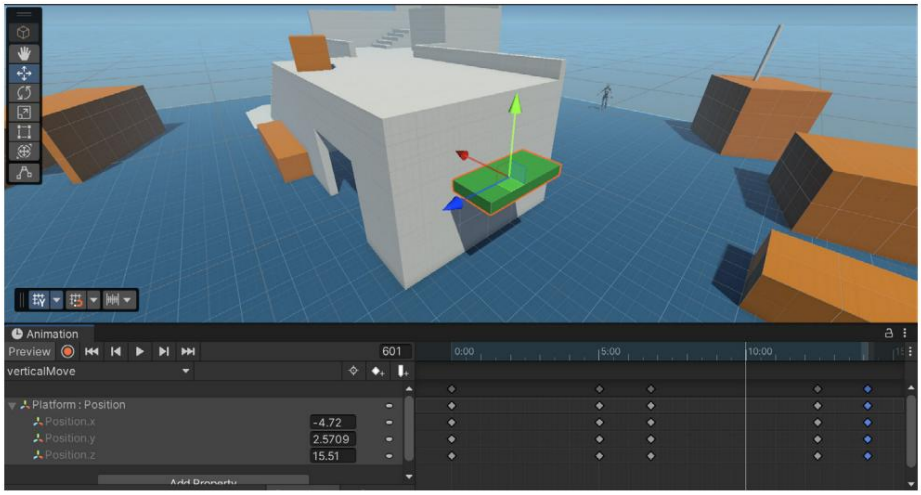
Animación



El sistema de animación

Las animaciones de personajes en Unity suelen crearse con software DCC como Blender, Autodesk Maya o Autodesk 3ds Max. El sistema de animación de Unity proporciona herramientas para modificar, refinar, adaptar proceduralmente y combinar dichas animaciones para darles vida en el juego o la experiencia interactiva que estás creando.

Sin embargo, en los prototipos de diseño de niveles, los elementos móviles no requieren sistemas de rigging complejos ni secuencias predefinidas. Las herramientas de animación de Unity te permitirán crear animaciones o secuencias aptas para prototipado, que podrán ser completamente implementadas y modificadas por animadores dedicados de tu equipo una vez que el diseño de niveles esté definido.

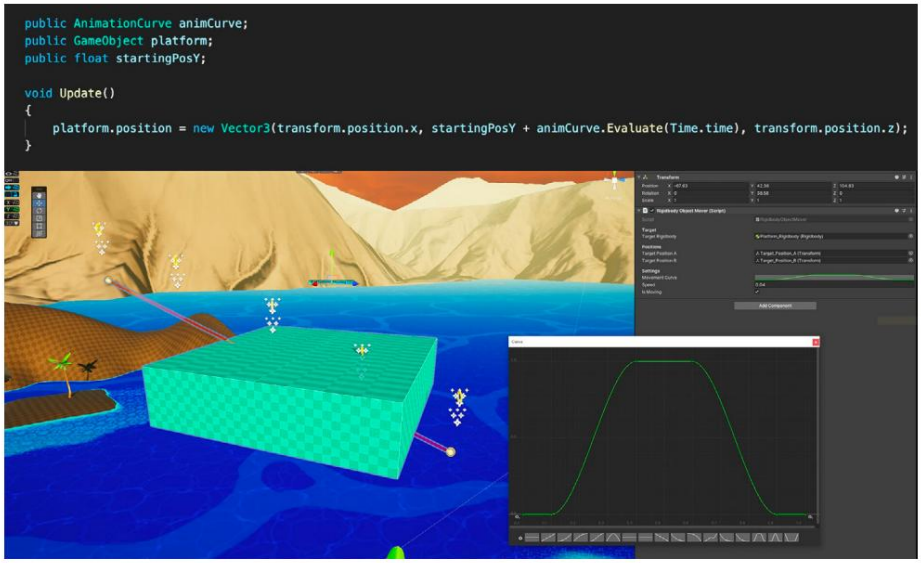


Animar una plataforma vertical con la ventana de Animación de Unity

El sistema de animación de Unity se basa en el concepto de clips de animación, que contienen información sobre cómo ciertos objetos deben cambiar su posición, rotación u otras propiedades con el tiempo. Cada clip puede considerarse una grabación lineal. Si su objeto tiene diferentes estados de animación, por ejemplo, "inactivo", "desactivado" o "activo", puede usar un controlador de animación, que es una máquina de estados de animación.

© 2023 Unity Technologies

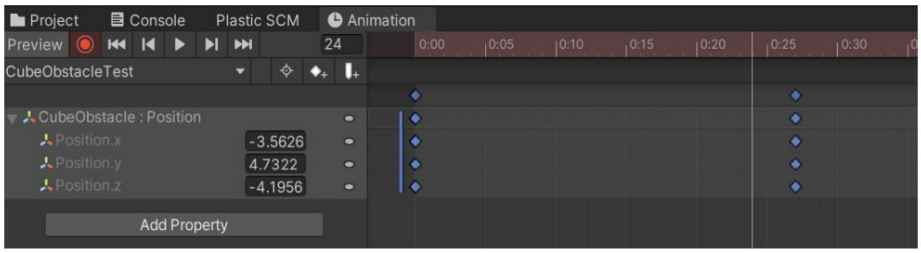
Machine Translated by Google



Puedes animar elementos programáticamente con la ayuda de curvas de animación, como se muestra en este blog. explica.

Ventana de animación

La ventana Animación permite crear y modificar clips de animación directamente en Unity. Está diseñada como alternativa a software de animación 3D externo o para crear animaciones sencillas según sea necesario durante el desarrollo. Proporciona el conjunto estándar de herramientas necesarias para la animación, como fotogramas clave, cabezal de reproducción, línea de tiempo de animación y curvas de animación.



En la ventana Animación, puedes cambiar fácilmente todos los fotogramas clave de la misma marca de tiempo seleccionando el punto en la parte superior. Puedes cambiar la duración de la animación seleccionando todos los fotogramas clave y moviendo los controladores a la izquierda y a la derecha.

Además del movimiento, puedes animar variables de materiales y componentes (casi cualquier propiedad de GameObject) y aumentar tus clips de animación con eventos de animación, que son funciones llamadas en puntos específicos a lo largo de la línea de tiempo, en el Editor.

La ventana Animación de Unity también te permite animar:

— La posición, rotación y escala de GameObjects

— Propiedades de los componentes, incluido el color del material y la intensidad de la luz. y volumen del sonido

— Propiedades dentro de sus scripts, como Float, Integer, Enumeración, vector y booleano

— El momento de llamar a funciones dentro de sus propios scripts

© 2023 Unity Technologies

Machine Translated by Google

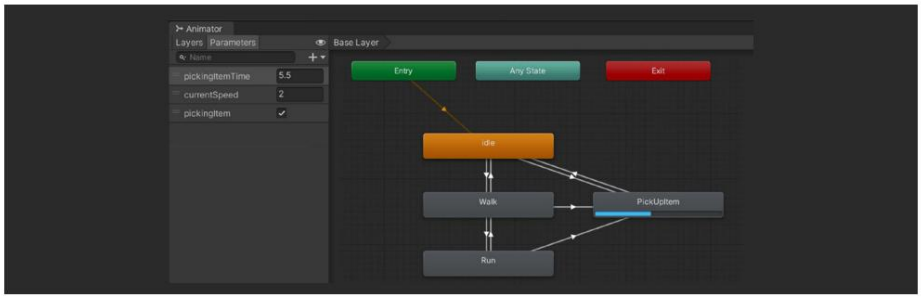
Los clips de animación generados se pueden utilizar mediante el controlador de animación o el aparejo de animación. kit de herramientas y aprovechamiento durante el juego o las cinemáticas con Timeline.

Utilice herramientas de animación como fotogramas clave y curvas para su proyecto.

Máquina de estados de animación

En Unity, el controlador de animación le permite organizar y mantener un conjunto de clips de animación y transiciones de animación asociadas para un personaje u objeto.

En la mayoría de los casos, es normal tener varias animaciones y alternar entre ellas cuando se dan ciertas condiciones del juego. Por ejemplo, se podría cambiar de un clip de caminar a uno de saltar al presionar la barra espaciadora. El controlador de animación tiene referencias a los clips de animación utilizados y los gestiona, así como las transiciones entre ellos, mediante una máquina de estados de animación. La máquina de estados puede considerarse como un diagrama de flujo de clips y transiciones, o como un programa simple escrito en un lenguaje de programación visual dentro de Unity.



Las ideas sobre el comportamiento de GameObjects se pueden probar visualmente con las máquinas de estados del controlador de animación.

Los árboles de mezcla son eficaces para ocultar la complejidad. Un árbol de mezcla No tiene estado ni se devuelve al código. Simplemente combina los diferentes clips según los parámetros definidos. Esto es importante porque permite iterar sobre árboles de mezcla sin preocuparse por dañar el resto del juego.
Incluso puedes ocultar una red compleja de estados para evitar errores en el futuro, ya que no puedes vincular el comportamiento a la mayoría de las animaciones en un árbol de mezcla.

Unity ofrece capas de animación Para gestionar máquinas de estados complejas. Por ejemplo, use capas de animación para crear una capa de la parte inferior del cuerpo para caminar y saltar, y una capa de la parte superior del cuerpo para lanzar objetos y disparar.

© 2023 Unity Technologies

Machine Translated by Google

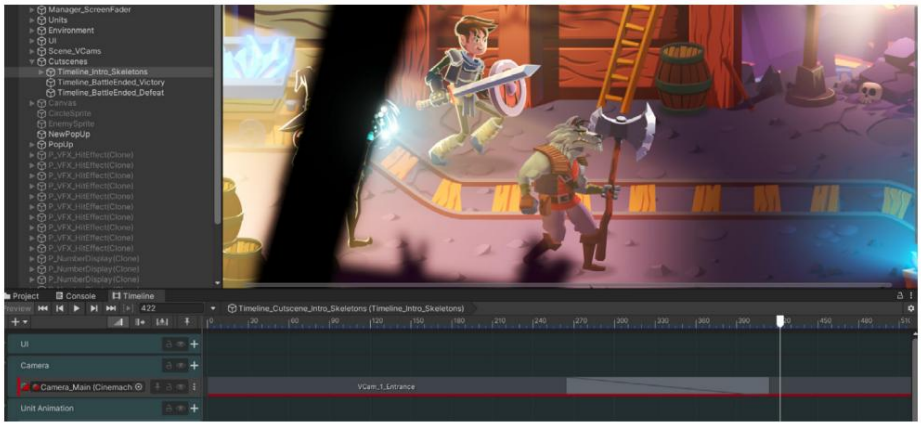
Además de la animación visual, los estados de animación pueden activar efectos de sonido o código C#.

Cronología

Cronología le permite orquestar los elementos de su escena, incluido el comportamiento, las animaciones y la habilitación o deshabilitación de objetos, así como crear secuencias o escenas de corte.

Al igual que en la edición de video no lineal, cada capa de la interfaz de la línea de tiempo es una pista. Al combinar varias pistas, puedes crear una función cinematográfica compuesta por audio, secuencias de juego, ángulos de cámara, efectos de partículas y más.

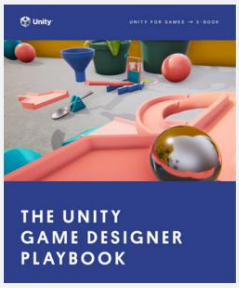
Si una secuencia programada es clave para tu juego o un momento específico del nivel, puedes activar una secuencia de eventos, como revelar una nueva puerta, despejar un camino, mostrar nuevos edificios o plataformas, etc. Considera usar la línea de tiempo para activar la secuencia llamando al componente PlayableDirector en un GameObject, como en este artículo de la comunidad. espectáculos.



Una escena de la demostración de Unity 2D Dragon Crashers siendo orquestada en Timeline

El manual del diseñador de juegos de Unity

Esta guía presenta las herramientas esenciales para que los diseñadores prototipen, creen y perfeccionen la jugabilidad en Unity. Obtén una introducción detallada a la programación en C# y a Unity Visual Scripting, y aprende a realizar tareas de diseño de juegos con una codificación mínima mediante control de entrada, controladores de personajes, caja gris y mucho más.



Descargar el libro electrónico

Más recursos

— Trabajar con clips de animación

— Mejore su flujo de trabajo con Animation Rigging

© 2023 Unity Technologies

Machine Translated by Google

— Reutilización y reorientación de animaciones entre plataformas

— Combinando jugabilidad y narrativa con Timeline: Cutscenes y gráficos del juego

La tienda de activos de Unity

La tienda de activos de Unity Contiene miles de recursos listos para usar, herramientas de producción, sistemas de juego y plantillas. Es un recurso valioso que ahorra tiempo, especialmente durante las primeras etapas del juego, el diseño de niveles y la preproducción, cuando necesitas probar y repetir rápidamente las ideas fundamentales del juego sin niveles, herramientas de diseño de niveles, arte ni mecánicas de juego completamente desarrollados. Si eres nuevo en Asset Store, comienza con esta guía.

Un collage de activos de Unity Asset Store

La Tienda de Activos está categorizada de forma que facilita la navegación y la búsqueda de activos durante la preproducción. Algunas sugerencias para empezar:

— Las plantillas Esta sección ofrece plantillas de juego completas con instrucciones detalladas para su personalización. Usa una plantilla como atajo para crear prototipos más detallados que puedas modificar a medida que avanzas. Por ejemplo, puedes configurar un bucle de juego completo para probar mientras desarrollas los pilares principales de tu juego.

— Muchos materiales diferentes Hay texturas disponibles para proyectos 3D y 2D. Usa materiales prefabricados para identificar diferentes superficies, ya sea para mostrar la jugabilidad o para crear accesorios y entornos más atractivos, al diseñar un nivel.

— Cambia tus formas primitivas por activos en 3D o 2D Sección de recursos que imita la apariencia de las versiones finales. Todo está disponible, desde entornos hasta personajes y accesorios. Además, se pueden visualizar... o los efectos de audio pueden ayudarte a transmitir el estado de ánimo que buscas.

Es difícil comunicar la narrativa o la intención del juego cuando solo tienes una cápsula que se mueve en un nivel prototipo rápidamente definido. En la sección de animación, encuentra conjuntos de animaciones con personajes ficticios que te ayudarán a crear una representación más precisa del diseño previsto.

© 2023 Unity Technologies

Machine Translated by Google

PARTE III : NIVEL

HERRAMIENTAS DE DISEÑO

UNIDAD

Cada juego tiene necesidades específicas en cuanto a la construcción de sus niveles. Un rompecabezas de combinar 3...
El juego podría requerir herramientas de depuración y diseño de niveles personalizados que permitan
Como diseñador de niveles, usted debe centrarse en ensamblar niveles con los muchos elementos móviles.
Piezas que entran en juego. Otros juegos pueden requerir muchas herramientas.
juntos para crear un gran entorno inmersivo al aire libre.

Unity proporciona muchas funciones listas para usar para diseñar niveles y para más
necesidades personalizadas, Unity Asset Store proporciona muchas herramientas de producción. Esto
Esta sección analiza las principales herramientas de Unity y una selección de recursos disponibles en
La tienda de activos.

Activos iniciales

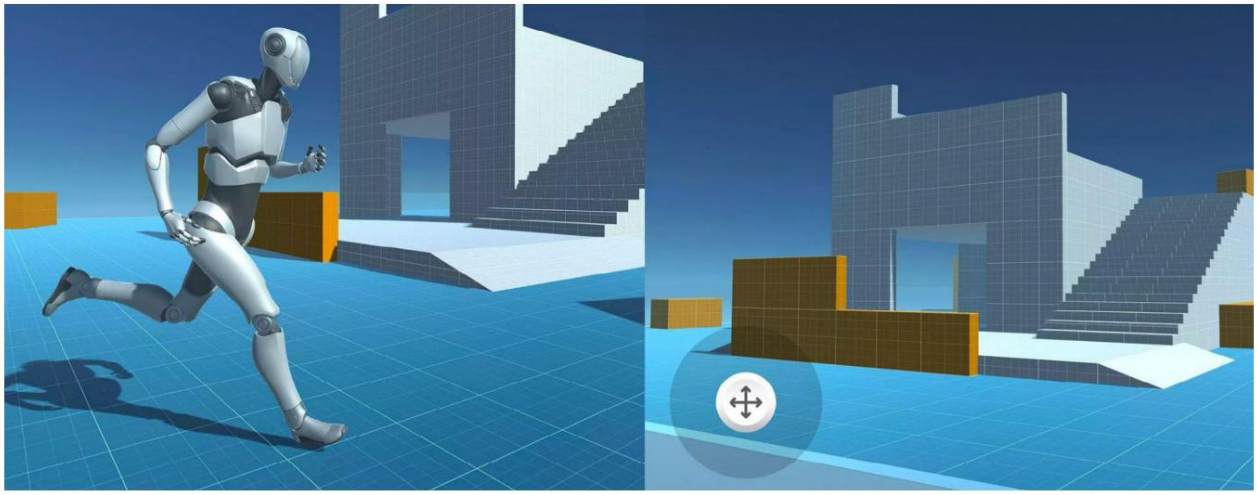
Una forma valiosa de probar nuevos entornos es navegarlos con un personaje jugable.
personaje. Los recursos iniciales son recursos gratuitos, proporcionados por Unity y disponibles en el
Unity Asset Store, que son útiles para crear prototipos rápidos con un personaje jugable
en cámaras de primera y tercera persona . Estos paquetes ofrecen controladores integrados.
forma modular que sirva como base sólida para cualquier género de juego.

Los recursos iniciales hacen uso del controlador de personaje componente y entrada
Sistema para mover al personaje, como hacerlo correr, saltar o mirar a su alrededor.
También hace uso de Cinemachine, El potente sistema de cámara de Unity, analizado
más adelante en este libro electrónico.

© 2023 Unity Technologies

Machine Translated by Google

Consigue un personaje configurado y listo para jugar.

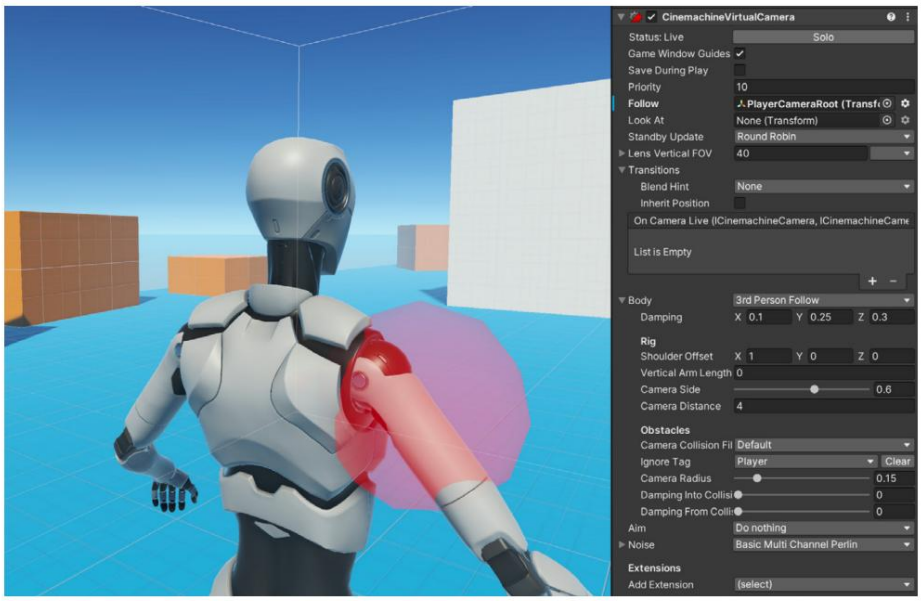


Los paquetes de recursos en tercera y primera persona incluyen controles para todas las plataformas. Selecciona la escena "Patio de recreo" y pruébalos ahora mismo.

Si importó los recursos de inicio en un proyecto y los materiales aparecen en magenta, asegúrese de actualizarlos a la canalización de renderizado universal (URP) o a la canalización de renderizado de alta definición (HDRP). Puede hacerlo en la carpeta Entorno/Arte/Materiales. También puede seleccionar los materiales en la carpeta Materiales y, en Ventana > Renderizado > Conversor de canalización de renderizado, marcar la opción Actualización de materiales y luego seleccionar Inicializar conversores > Convertir recursos.

© 2023 Unity Technologies

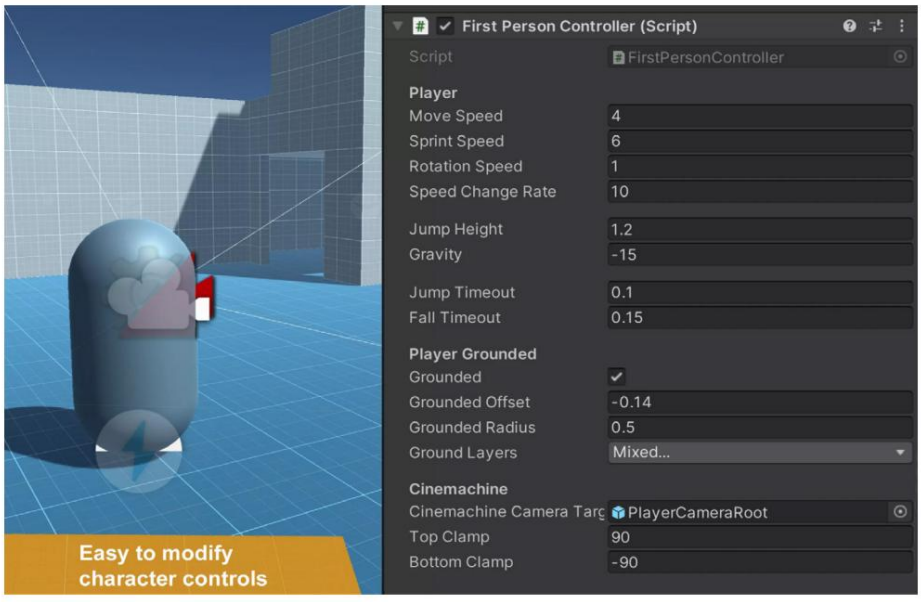
Machine Translated by Google



El paquete de activos en tercera persona utiliza una cámara Cinemachine con configuraciones diseñadas para un seguimiento en tercera persona, como opciones para evitar obstáculos de la cámara.

Puedes tener ambos recursos en el mismo proyecto si quieres probar tus niveles en primera o tercera persona. Asegúrate de incluir tu nivel en ambas escenas (llamado Playground) o, alternativamente, incluye los prefabricados de personaje, control y cámara en tus escenas.

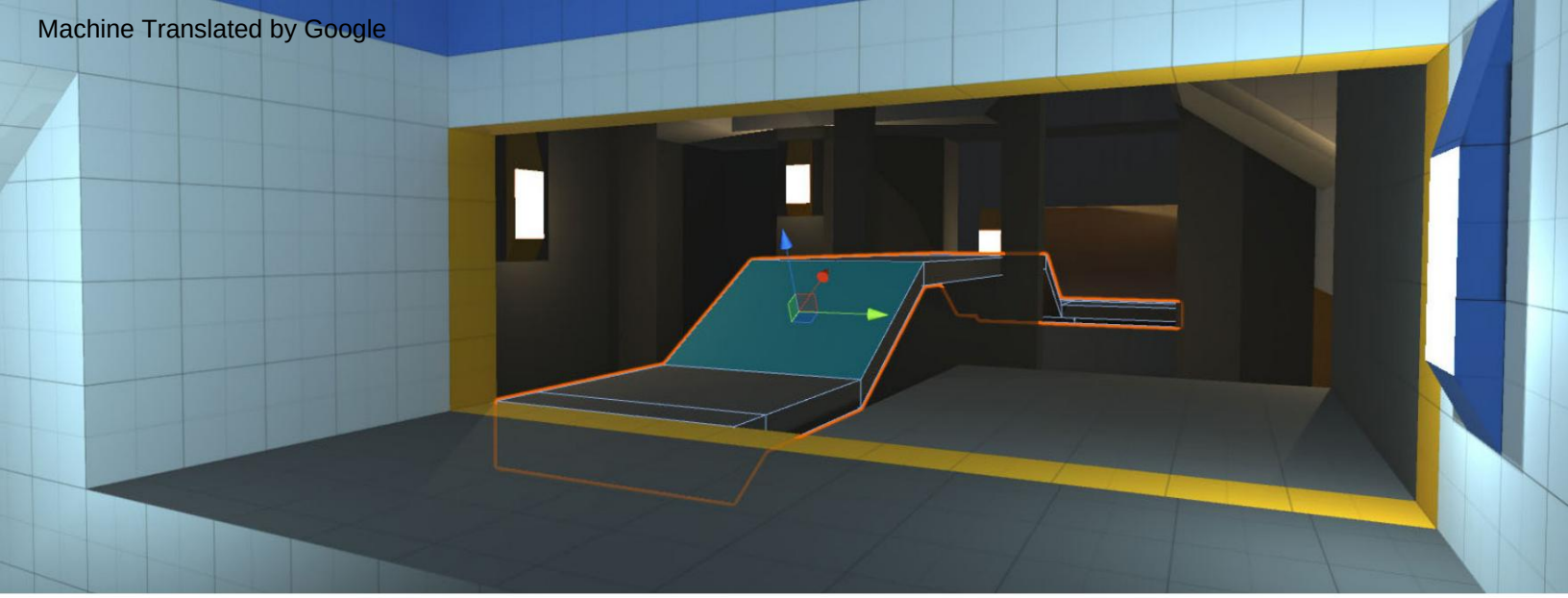
Cada activo de inicio viene con documentación ubicada en StarterAssets carpeta. Busque el recurso Scene Playground en la carpeta Project y ábralo para comenzar a usar los recursos.



El script Controlador de Primera y Tercera Persona expone valores que permiten ajustar el movimiento del personaje. Utiliza el Controlador de Personaje, que no reacciona a las fuerzas por sí solo ni empuja automáticamente los componentes del Rigidbody a menos que se indique en un script como el script BasicRigidBodyPush adjunto.

© 2023 Unity Technologies

Machine Translated by Google

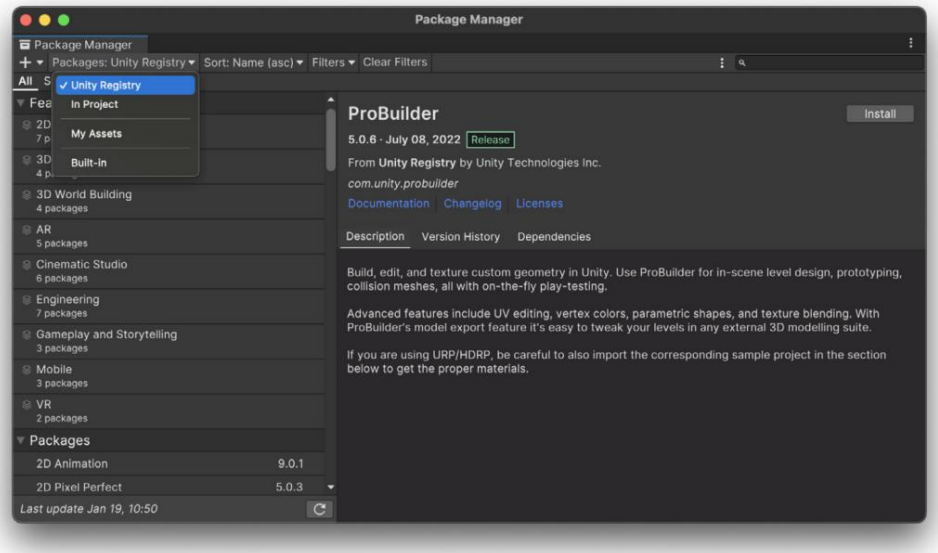


Comience a diseñar niveles en Unity con la herramienta de modelado 3D incorporada ProBuilder.

ProBuilder

ProBuilder y Polybrush son herramientas que le permiten diseñar, en caja gris, Prototipo y niveles de prueba en Unity sin software de modelado 3D o Artistas 3D profesionales. Modela tu nivel con ProBuilder y luego añade detalles con Polybrush como texturas de pintura, esculpir la malla existente, etc.

Incluso puedes lanzar tu juego con niveles creados en ProBuilder, como el equipo detrás de SUPERHOT hizo.



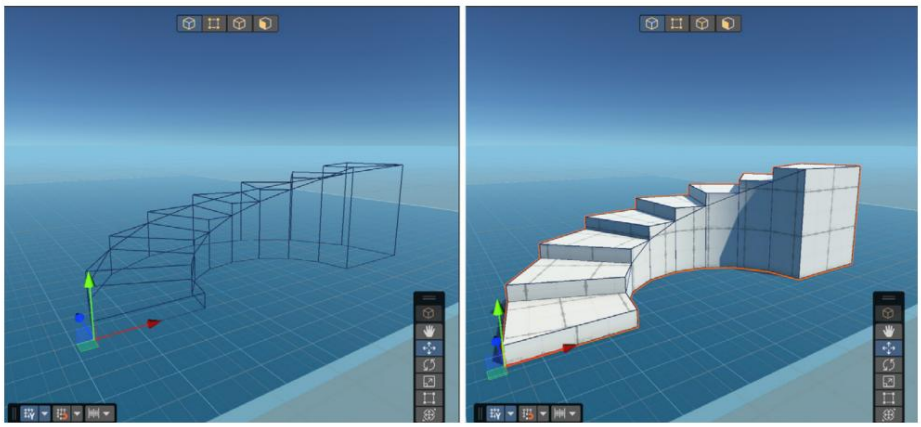
Recuerde instalar los archivos de soporte para URP o HDRP si su proyecto utiliza esas canalizaciones de renderizado

Al instalar el paquete ProBuilder, asegúrese de importar los archivos de soporte (materiales) que corresponden a URP o HDRP. De lo contrario, los objetos podrían

No se renderizará. Si utiliza la canalización de renderizado integrada, no es necesario realizar ninguna acción.

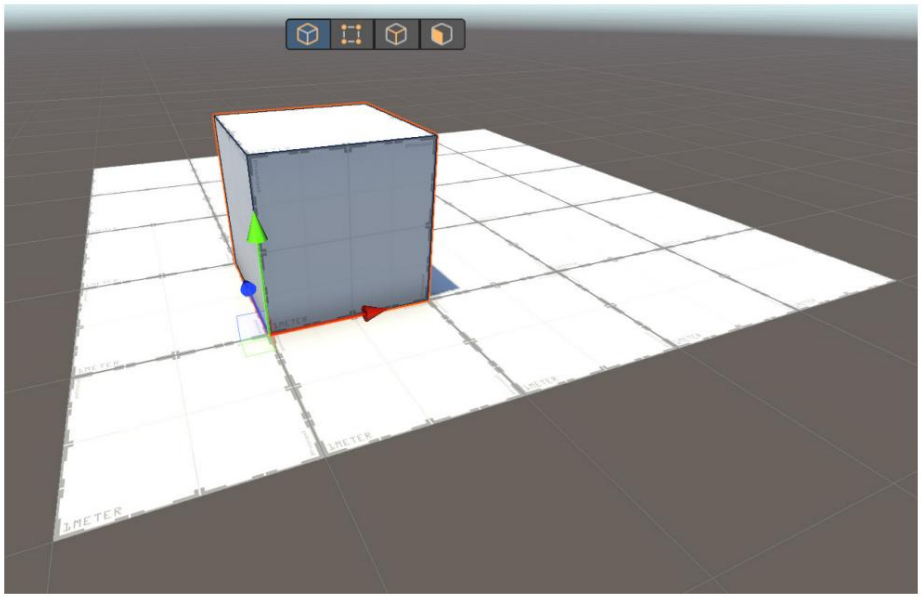
© 2023 Unity Technologies

Machine Translated by Google



Un nuevo proyecto en Unity 2022 LTS usando URP, una escalera creada antes y después de instalar el archivo de soporte

Cree un objeto primitivo ProBuilder desde el menú superior a través de GameObject > ProBuilder > Cubo (por ejemplo). ProBuilder incluye varias primitivas Formas útiles para la creación rápida de prototipos. Estas formas, que también vienen con un componente Mesh Collider para física, se puede agregar en la vista de Escena.

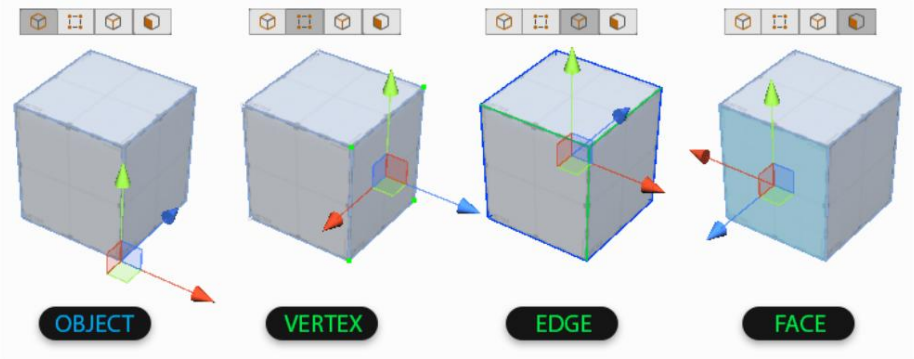


Los primitivos de ProBuilder utilizan un material con una textura que tiene un patrón de cuadrícula de una unidad Unity (equivalente a un metro de el mundo real). Esto le ayuda a realizar un seguimiento de la escala del objeto y permite replicar en una escena el tamaño de cualquier espacio dado en el mundo real.

La ventana principal para trabajar con ProBuilder se encuentra en Herramientas > ProBuilder > Ventana de ProBuilder. Al habilitarlo, también se agregará una barra de herramientas en la Escena. Ver para editar mallas de ProBuilder.

© 2023 Unity Technologies

Machine Translated by Google



Hay cuatro formas de manipular mallas en ProBuilder:

— Modo objeto: el modo estándar de Unity para seleccionar y manipular
Objetos del juego

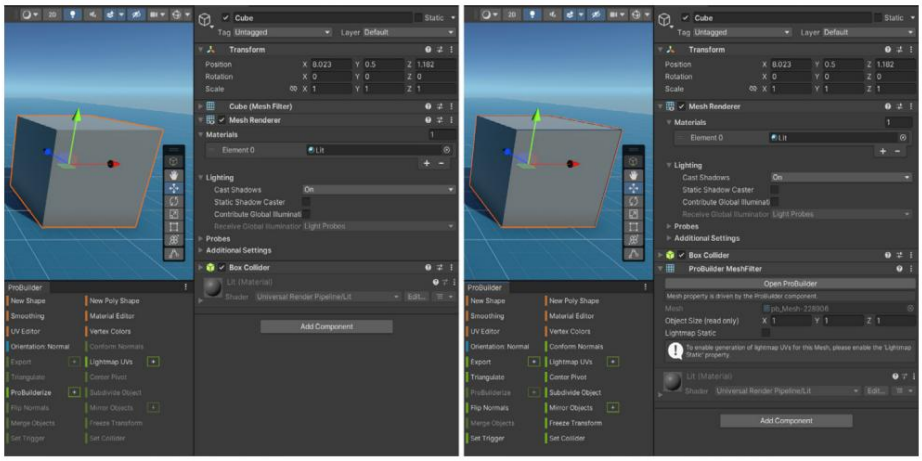
— Modo vértice: El modo de elemento para seleccionar y manipular vértices (puntos)

— Modo de borde: El modo de elemento para seleccionar y manipular bordes (líneas)

— Modo cara: El modo de elemento para seleccionar y manipular caras (polígonos)

Considere configurar accesos directos para la acción de ProBuilder en el menú de accesos directos para cambiar rápidamente entre los modos de manipulación.

Las mallas de ProBuilder funcionan como GameObjects normales en Unity. Puedes aplicar valores de transformación, añadir componentes, física y scripts, y animarlos.
Sin embargo, las mallas estándar de Unity no son lo mismo que las mallas de ProBuilder: no puedes editarlas con ProBuilder hasta que las conviertas en objetos de ProBuilder.

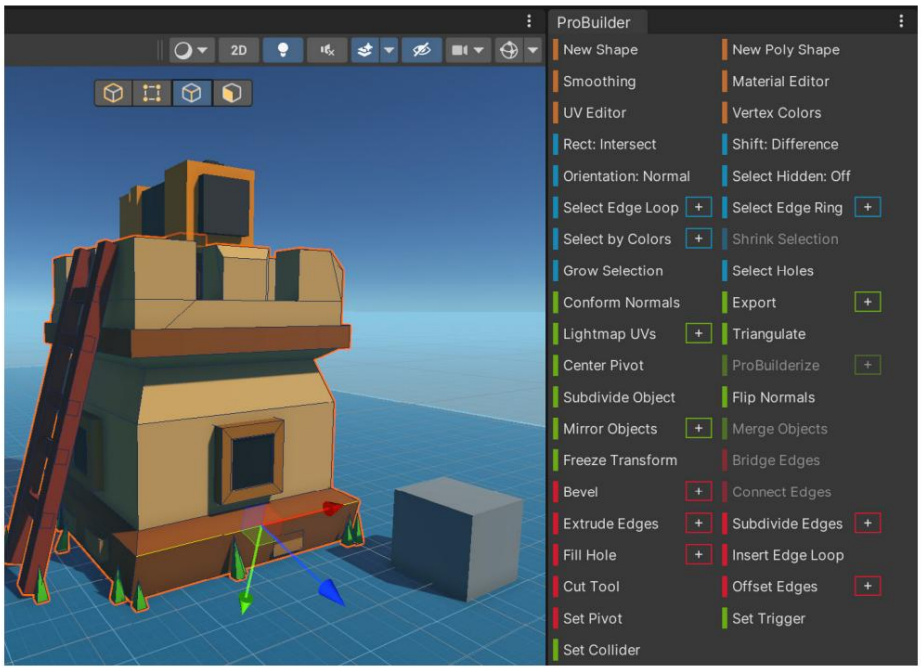


Un cubo de GameObject antes y después de convertirse en una malla ProBuilder con ProBuilderize acción

© 2023 Unity Technologies

Machine Translated by Google

Si bien es común construir una malla ProBuilder creándola y editándola completamente con herramientas y acciones de ProBuilder, también puede ser útil modificar activos existentes; por ejemplo, puede importar activos desde Unity Asset Store y modificarlos con ProBuilder.



Un activo de torre importado de Unity Asset Store; cuando lo convierte a través de la acción ProBuilderize, se puede modificar como una malla ProBuilder

La ventana principal de ProBuilder está codificada por colores para ayudarle a elegir herramientas por tipo:

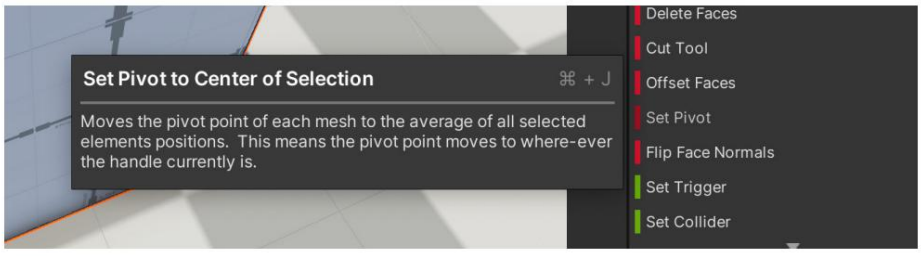
— Naranja para paneles de herramientas (ventanas que se abren por separado)

— Azul para funciones para configurar y cambiar la selección

— Verde para acciones de edición de malla que afectan a todo el objeto

— Rojo para funciones de edición de malla que actúan sobre vértices, bordes y Geometría facial

ProBuilder también incluye información sobre herramientas para acceder a información adicional sobre las funciones que se pueden ver manteniendo el mouse sobre una herramienta.

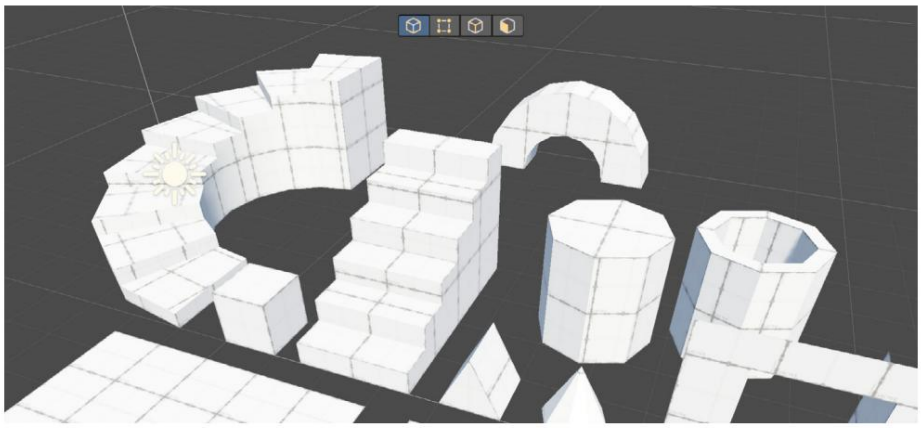


Aparece una información sobre herramientas al pasar el cursor sobre una de las herramientas de ProBuilder; un símbolo + junto a una herramienta indica opciones adicionales

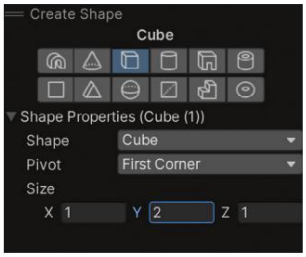
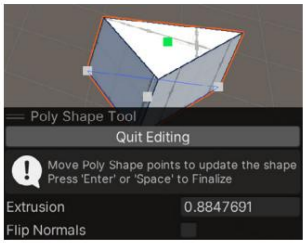
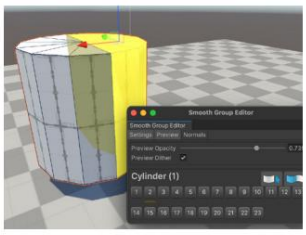
© 2023 Unity Technologies

Machine Translated by Google

Las siguientes tablas proporcionan información sobre lo que hace cada herramienta en ProBuilder.
Muchas de estas herramientas tienen un signo + al lado que indica que vienen con
Opciones ajustables. Algunas de las herramientas enumeradas aquí se describen con mayor detalle.
más adelante en esta sección.

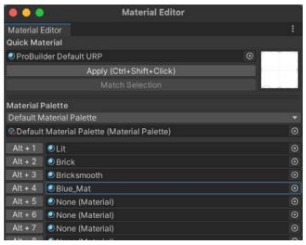
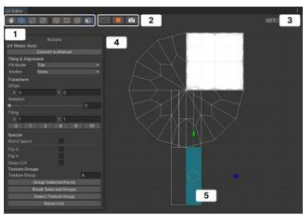
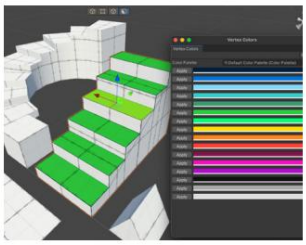


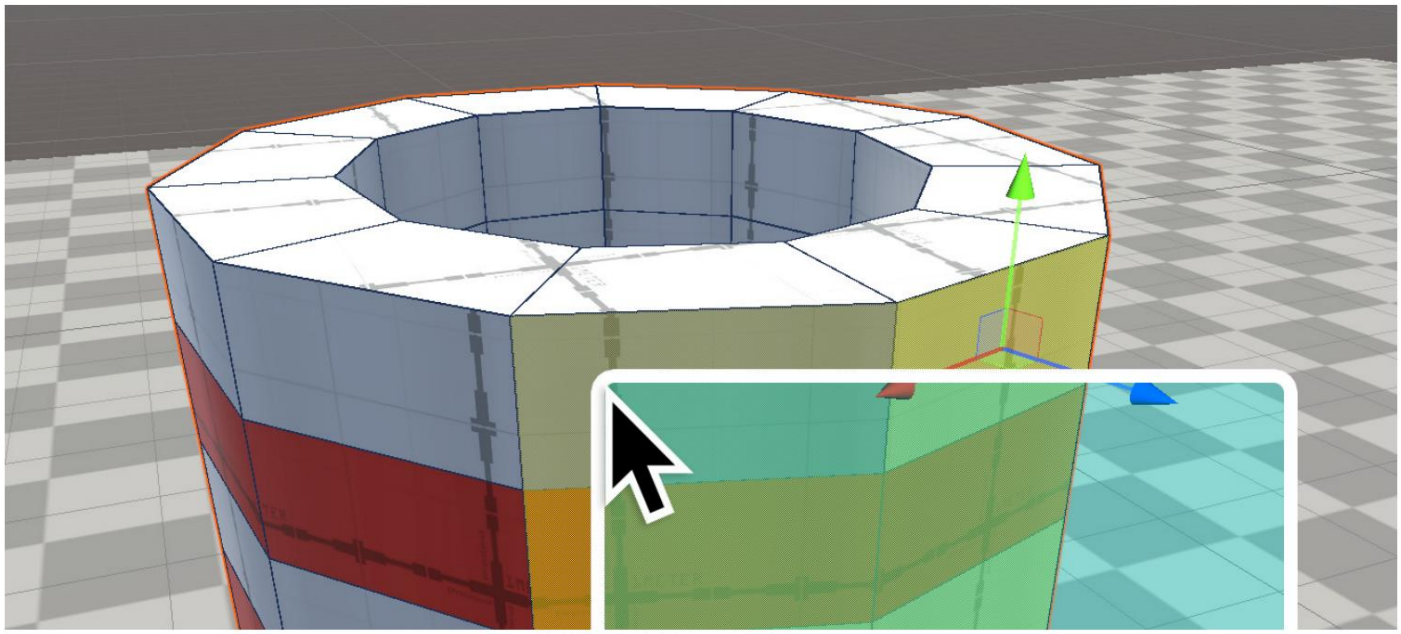
Algunas de las formas predeterminadas incluidas en ProBuilder

Color	Nombre	Imagen de referencia	Descripción
	Nueva forma		— La herramienta más común para creando niveles en ProBuilder — Seleccione una de las formas, Luego seleccione el preferido posición de pivote y definir la tamaño ya sea en la vista de Escena o en esta ventana T — Opciones adicionales disponibles, basado en la forma, a ajustar la definición de la malla
	Nueva forma de poli		— Crea una forma 2D personalizada y luego extruirlo en una malla 3D — Bueno para construir rápidamente una estructura irregular — También es útil para crear un 3D mapa de un plano de dibujo
	Suavizado		— Suaviza grupos de polígonos para que formen una superficie lisa y uniforme — Lea más aquí

© 2023 Unity Technologies

Machine Translated by Google

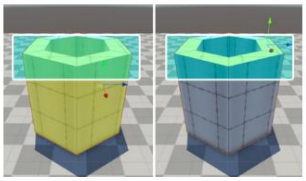
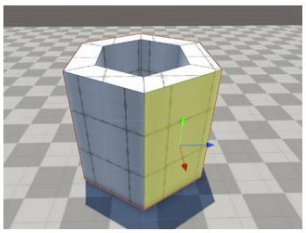
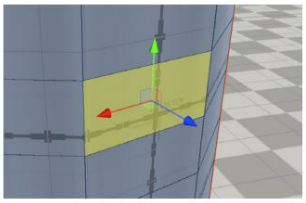
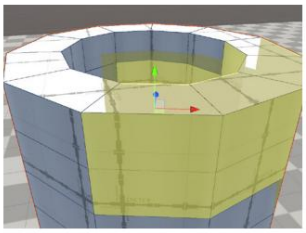
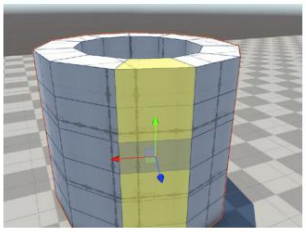
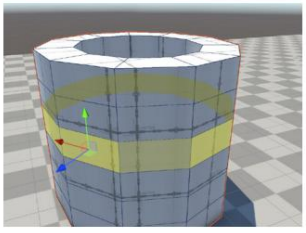
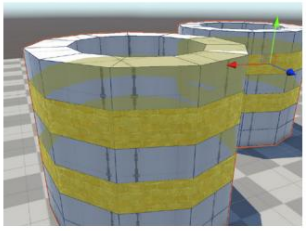
	Editor de materiales		— Aplica materiales a objetos o caras rápidamente desde una biblioteca definida de materiales — La aplicación de materiales en prototipos para indicar la intención o la función puede ayudar a transmitir ideas de diseño con mayor claridad.
	Editor UV		—Similar al suavizado característica; se utiliza mejor para prototipos más refinados o diseño de niveles — Lea más aquí
	Colores de vértice		— Equivalente del Material Editor pero sin textura modelos — Aplique color a caras, vértices o bordes para agregar más información a sus prototipos de diseño de niveles .



Es necesario haber seleccionado un elemento de la malla, como una cara, para habilitar algunas de las funciones.

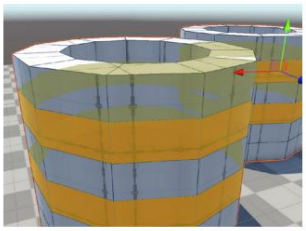
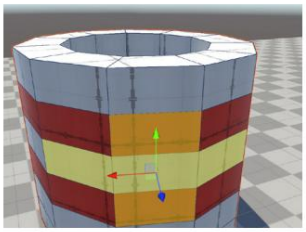
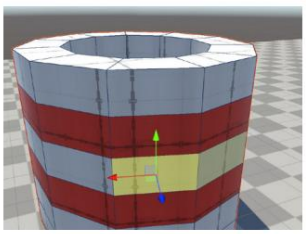
© 2023 Unity Technologies

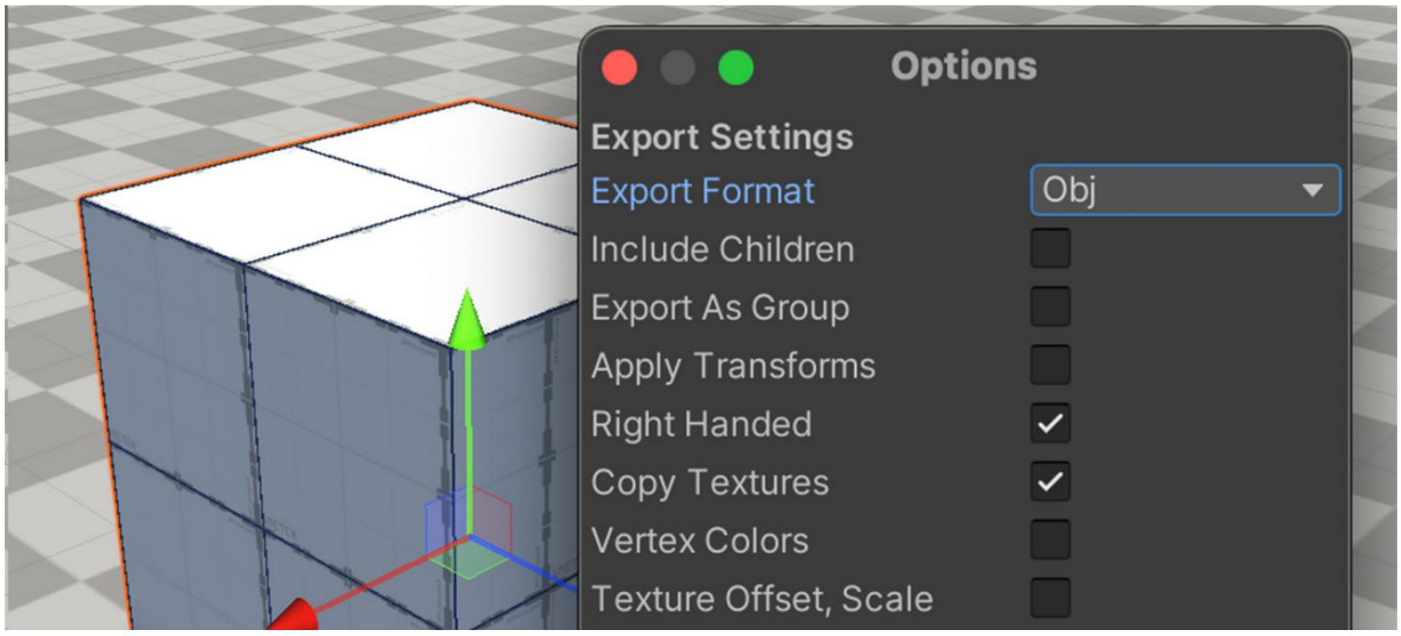
Machine Translated by Google

Color	Nombre	Imagen de referencia	Descripción
	Rectángulo		— Úselo para elegir si la selección de arrastre solo debe seleccionar elementos completamente dentro del rectángulo de arrastre, o también elementos parcialmente dentro del rectángulo de arrastre
	Cambio		— Cambiar lo que sucede con el selección de vértices, caras o aristas al arrastrar y seleccionar — Agregar, restar o invertir los elementos seleccionados
	Orientación		— Establece la orientación de los controladores al seleccionar objetos y elementos — Global sigue la orientación mundial, Local es relativo a la rotación del objeto y Normal sigue las normales de las caras o vértices.
	Seleccionar Oculto		— Selecciona o ignora elementos ocultos desde la cámara, como los que están en la parte posterior de la malla cuando arrastras y seleccionas
	Seleccionar cara Bucle		— Cuando se selecciona una cara, se expande la selección a todas las caras continuas. caras verticalmente
	Seleccionar cara Anillo		— Cuando se selecciona una cara, ampliar la selección a todas las caras continuas horizontalmente
	Seleccionar por Material		— Selecciona todas las caras en la escena que tienen el mismo material que el cara seleccionada — Limitar la selección a la actual objeto habilitando la Selección actual en el menú emergente

© 2023 Unity Technologies

Machine Translated by Google

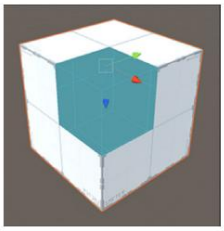
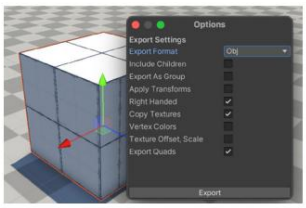
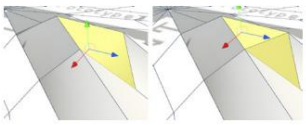
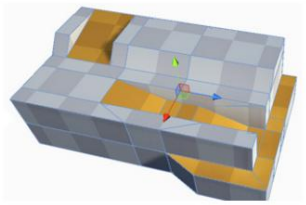
	Seleccionar por Bandera		— Funciona como Seleccionar por material pero utiliza colores de vértice
	Crecer Selección		— Expande la selección hacia afuera a caras, bordes o vértices adyacentes Limitar la selección a elementos dentro un ángulo especificado
	Encoger Selección		— Elimina elementos en el perímetro de la selección actual



Las funciones de edición de malla para un objeto ProBuilder están disponibles cuando selecciona el objeto o uno de sus elementos.

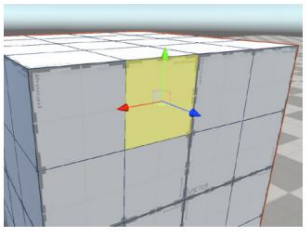
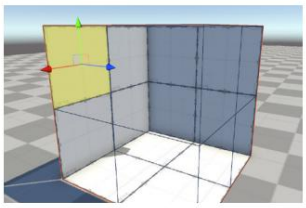
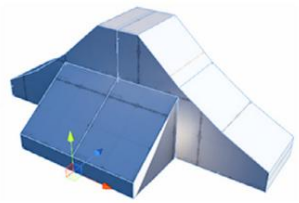
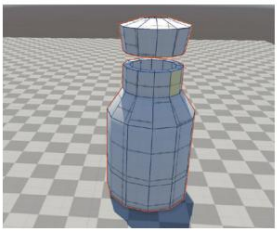
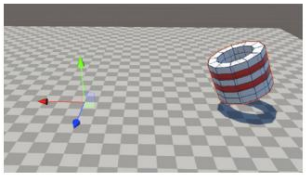
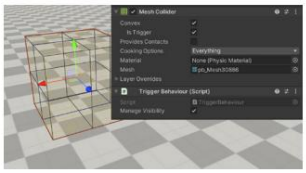
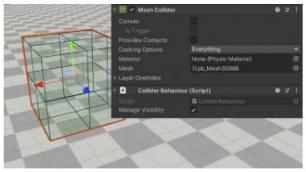
© 2023 Unity Technologies

Machine Translated by Google

Color	Nombre	Imagen de referencia	Descripción
	Ajustarse Normales		— Corrige desajustes en el triángulo devanado — Corregirá la dirección de la desviación. superficies orientadas en sentido contrario caras seleccionadas
	Exportar		— Exportar el objeto seleccionado a un Archivo de modelo 3D como OBJ, STL, Prefabricado PLY o Unity — Permite a los artistas tener una referencia de su prototipo 3D modelo en el DCC de su elección — Lea más aquí
	Mapas de luz UV		— De forma predeterminada, automáticamente actualiza los UV del mapa de luz mientras trabajando con la malla —Es posible desactivarlo y hacerlo. manualmente antes de hornear las luces
	Triangular		— Divide las caras en su base. triángulos para crear un facetado, apariencia no lisa — Utilice grupos de suavizado para volver a aplicar suavizar si es necesario
	Pivote central		— Mueve el punto de pivote para el Malla al centro de la límites del objeto
	ProBuilderize		— Convierte objetos 3D seleccionados en versiones editables en ProBuilder — Útil para modificar los existentes activos y agregarlos a un prototipo de nivel

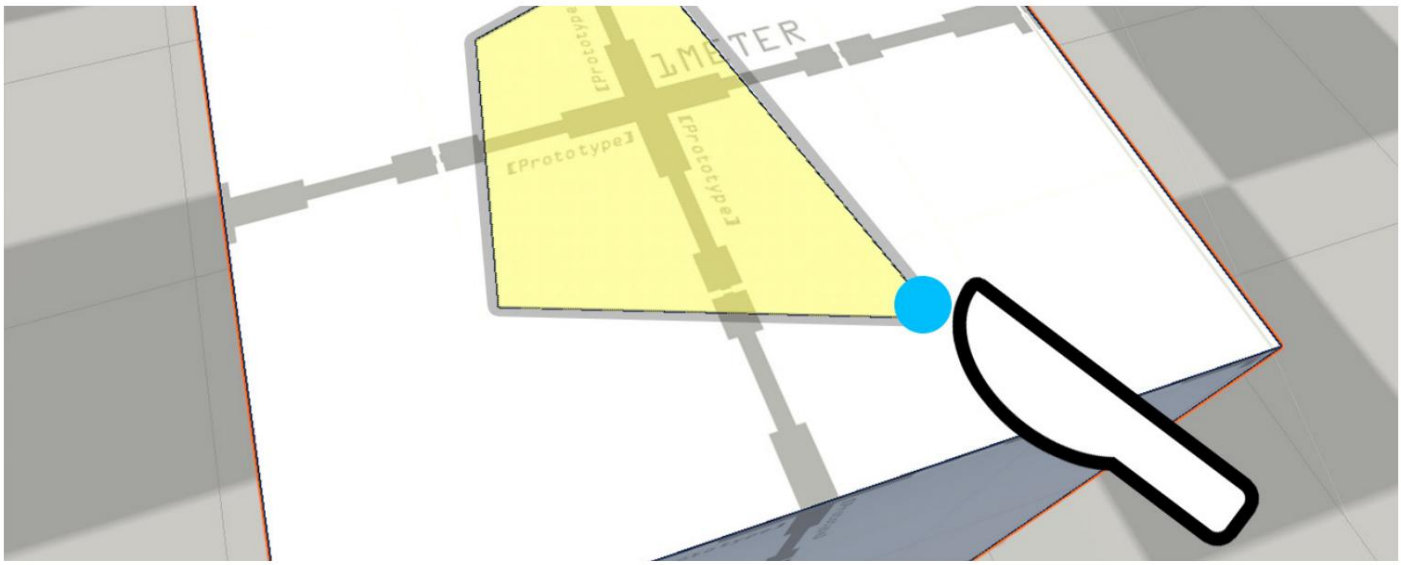
© 2023 Unity Technologies

Machine Translated by Google

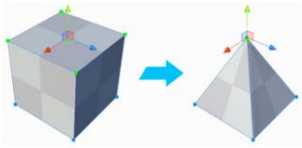
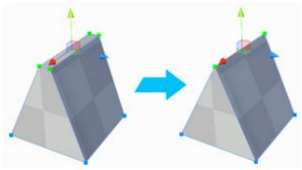
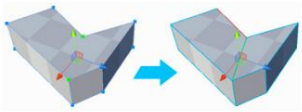
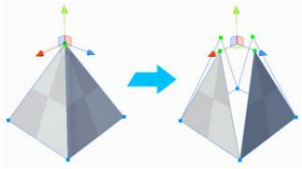
	Subdividir		— Divide cada cara en la zona seleccionada objetos, lo que permite una mayor niveles de detalle al modelar
	Invertir normales		— Invierte las normales en el elemento seleccionado. sólo objeto
	Objetos espejo		— Crea copias reflejadas de objetos, lo cual es útil cuando creando elementos simétricos
	Fusionar objetos		— Fusiona dos o más elementos seleccionados Objetos ProBuilder en uno
	Congelar Transformar		— Establece el punto de pivote del objeto en La posición actual del origen mundial (0,0,0) y restablece la rotación y escala
	Establecer disparador		— Asigna el comportamiento del disparador script para seleccionar objetos Conviértalo en un Trigger Collider en el juego
	Colisionador de conjuntos		— Asigna el comportamiento del colisionador script para seleccionar objetos Conviértalo en un colisionador en el juego

© 2023 Unity Technologies

Machine Translated by Google

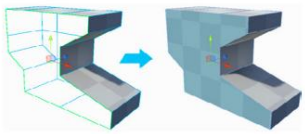
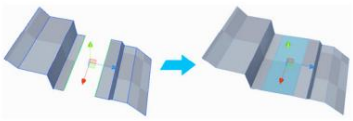
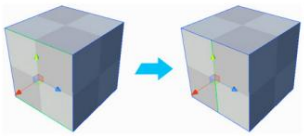
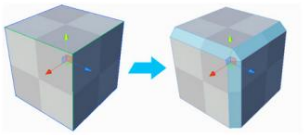
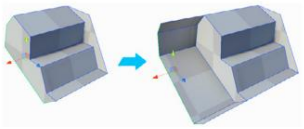
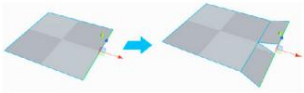
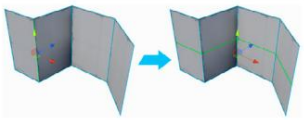


Funciones de edición de malla que actúan sobre la geometría de vértices, aristas y caras

Color	Nombre	Imagen de referencia	Descripción
Operaciones de vértice			
	Colapso de vértices		— Contrae todo lo seleccionado vértices ya sea a un centro punto o la posición de la primer vértice seleccionado
	Vértices de soldadura		— Fusiona vértices seleccionados dentro de una distancia específica de unos y otros — Ayuda a optimizar la geometría para vértices lo suficientemente cercanos a entre sí
	Conectar vértices		— Crea una nueva ventaja para conectar los vértices seleccionados
	Vértices divididos		— Divide un vértice en elementos individuales. vértices (uno por cada adyacente) cara) para que puedas moverla se enfrenta de forma independiente cuando seleccionas cada uno de ellos

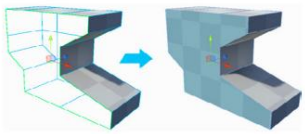
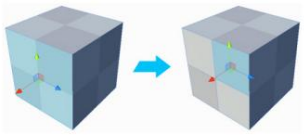
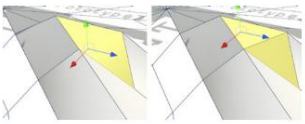
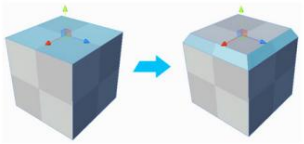
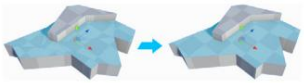
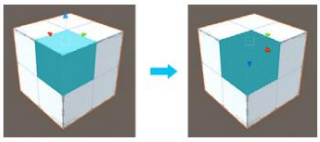
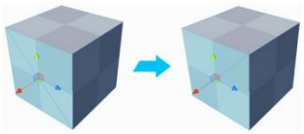
© 2023 Unity Technologies

Machine Translated by Google

	Rellenar agujero		— Crea una nueva cara que rellena cualquier agujero que toque el vértices seleccionados.
Operaciones de borde			
	Bordes del puente		— Crea una nueva cara entre dos bordes seleccionados
	Conectar bordes		— Inserta un borde que conecta los centros de cada seleccionado borde
	Bisel		— Divide los bordes seleccionados en dos, con una nueva cara entre ellos a ellos
	Extruir bordes		— Empuja un nuevo borde hacia afuera cada borde seleccionado, Conectados por una nueva cara para cada borde — Sólo funciona en bordes abiertos
	Subdividir bordes		— Divide los bordes seleccionados en múltiples bordes (dos por por defecto)
	Insertar bucle de borde		— Desde el borde seleccionado, agrega un nuevo bucle de borde, que es un serie de aristas directamente conectados a través de quads.

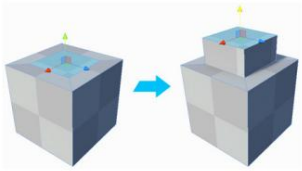
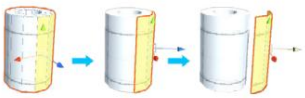
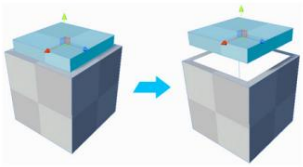
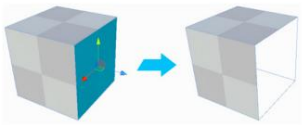
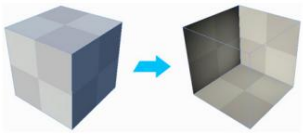
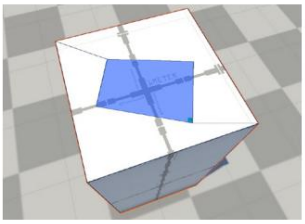
© 2023 Unity Technologies

Machine Translated by Google

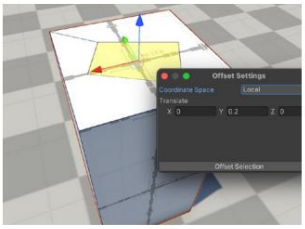
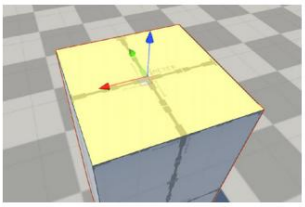
	Rellenar agujero (bordes)		— Crea una nueva cara que rellena cualquier agujero que toque el bordes seleccionados
Operaciones con caras poligonales			
	Subdividir caras		— Divide cada cara seleccionada, añade un vértice en el centro de cada borde y los conecta en el centro
	Triangular caras		— Divide la cara seleccionada en sus triángulos base para crear un facetado, no liso apariencia — Si es necesario, utilice suavizante. grupos para volver a aplicar el suavizado
	Caras biseladas		— Divide los bordes de la cara seleccionada en dos aristas, con una nueva cara entre
	Fusionar caras		— Fusiona las caras seleccionadas en una una sola cara y elimina cualquier bordes divisorios
	Normales conformes		— Corrige desajustes en el triángulo devanado — Corregirá la dirección de la desviación. superficies orientadas en sentido contrario en caras seleccionadas
	Borde de la cara giratoria		— Intercambia la orientación del triángulo en la(s) cara(s) seleccionada(s) con cuatro lados, lo que da como resultado una inversión de la dirección de la borde medio en un quad

© 2023 Unity Technologies

Machine Translated by Google

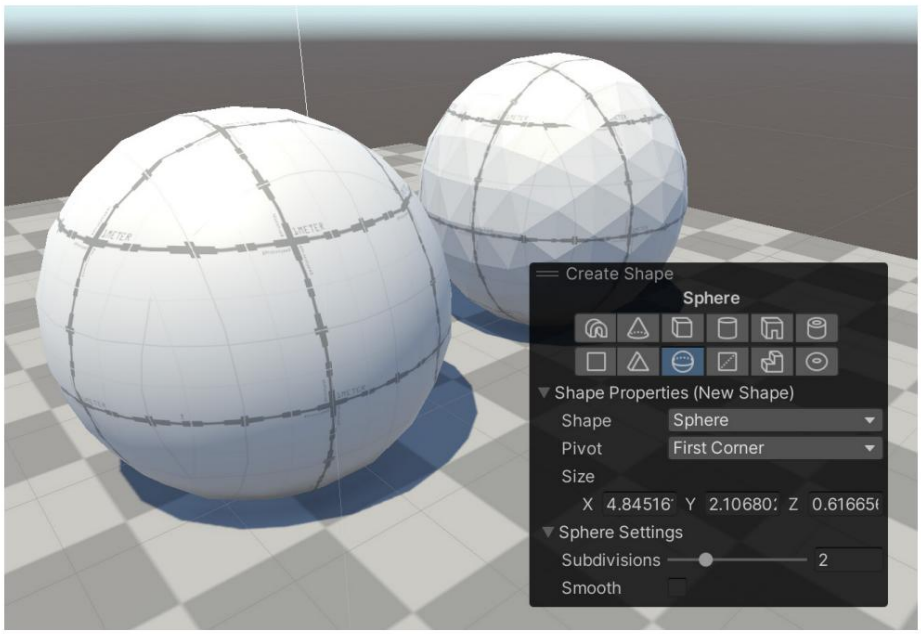
	Extruir caras		— Crea una nueva cara tirando fuera de la cara seleccionada actualmente y uniendo los lados a cada uno borde — De forma predeterminada, cada cara nueva sigue la dirección de su normales de vértice
	Caras duplicadas		— Copia cada cara seleccionada y crea una nueva malla
	Separar caras		— Separa las caras seleccionadas del resto de la malla
	Eliminar caras		— Elimina las caras seleccionadas
	Voltear normales de cara		— Voltea solo las normales en el caras seleccionadas, práctico práctico para construir habitaciones y Cambiar las paredes desde el exterior fachadas a interiores.
Operaciones comunes			
	Herramienta de corte		— Subdividir caras; definir la Forma recortada con puntos y se convierte en una nueva cara en el Yo

Machine Translated by Google

	Compensar		— Le permite ingresar un valor preciso para mover vértices, aristas y caras
	Establecer pivote		— Mueva el punto de pivote de este Malla al centro promedio de los vértices, aristas o caras seleccionados

Grupos de suavizado

Grupos de suavizado Sirven para crear bordes afilados o suaves entre caras. Cuando los polígonos vecinos no comparten el mismo grupo de suavizado, se crea un borde afilado entre ellos. Imaginemos los polígonos que componen el parabrisas o la ventana delantera de un coche: formarían un grupo de suavizado, mientras que los polígonos que componen el capó serían otro grupo. Si los polígonos del parabrisas y del capó forman parte de la misma malla, ambos se suavizarán, pero no se considerarán la misma superficie.



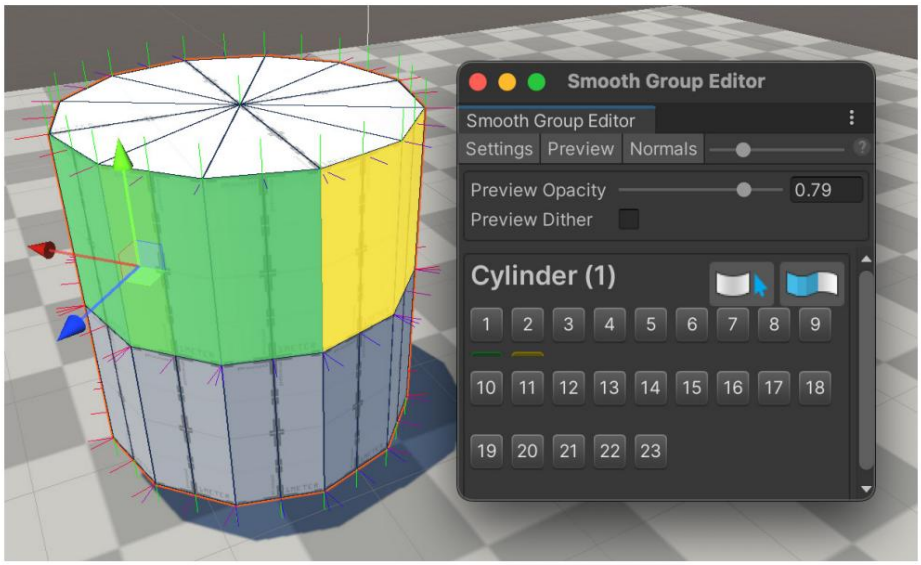
La segunda esfera tiene la opción Suavizar desactivada. Puedes cambiar esta configuración en el Inspector seleccionando el objeto.

© 2023 Unity Technologies

Machine Translated by Google

En ProBuilder, las formas nuevas tienen todas las caras suavizadas por defecto. Para mayor control sobre los grupos de suavizado, desactive la opción "Suavizado" (como se muestra en la imagen superior).

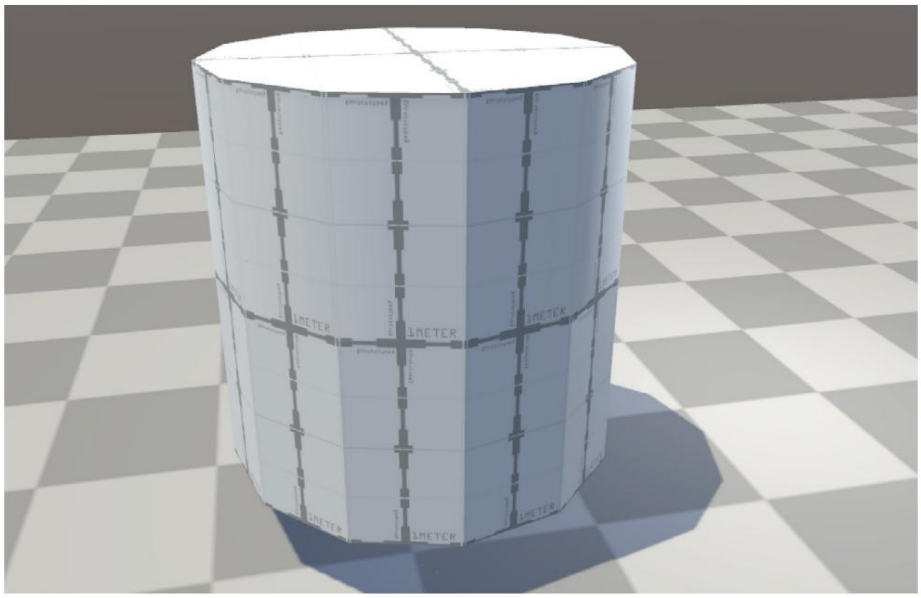
La ventana del editor de grupos Smooth ofrece opciones para ayudarle a configurar y previsualizar los diferentes grupos de suavizado.



Cuando la ventana Editor de grupos suaves está abierta, la malla mostrará los diferentes grupos mediante un código de color.

Con la opción Vista previa, puedes ajustar la transparencia del código de color de las caras suavizando el grupo y fusionándolas con un efecto de tramado. También puedes previsualizar las normales de los vértices.

El ejemplo anterior se ve así en la vista del Juego:



Observe cómo la parte superior del cilindro y su mitad inferior no tienen suavizado, lo que permite ver los bordes de las caras. Las dos partes superiores están suavizadas, pero se aprecia una costura en el centro. Esto se debe a que ambas partes pertenecen a grupos de suavizado diferentes y, por lo tanto, se tratan como superficies distintas.

© 2023 Unity Technologies

Machine Translated by Google

Veamos los pasos para configurar grupos de suavizado:

1. Seleccione las caras del objeto que pertenecerán al mismo grupo, de modo que serán tratados como la misma superficie.

2. Haga clic en uno de los botones del 1 al 23 en la ventana Editor de grupo suave, y las caras seleccionadas se asignarán a ese grupo. a. Aparecerá una línea fina de color debajo del botón en el que hizo clic. Esto indica que ahora se crea un grupo, con el color de la línea coincidiendo con el color de la previsualización para el grupo. Los rostros pertenecientes al grupo 2 en el

   Las imágenes de arriba son de color amarillo en la imagen de previsualización que se muestra más arriba, como Es la línea debajo del botón.

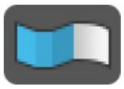
3. Seleccione otro conjunto de caras y haga clic en un nuevo botón para asignarlas a una nueva Agrupa con un color diferente. Repite los pasos tantas veces como sea necesario.

Hay dos opciones para seleccionar las caras pertenecientes a uno de los grupos:

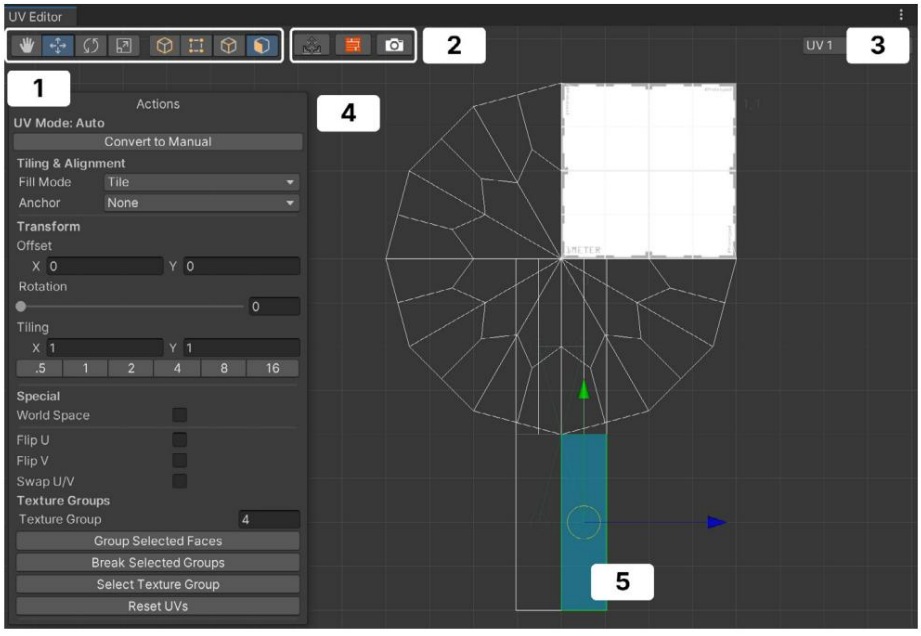
— Haga clic derecho en el botón del grupo que desea seleccionar.

— Seleccione una de las caras del objeto y haga clic en el botón con una flecha azul para expandir la selección a todas las caras pertenecientes a el mismo grupo.

Para eliminar un grupo, seleccione todas las caras del mismo grupo como mencionado anteriormente y luego haga clic en el botón con caras azules



Editor UV



Esta es la ventana del Editor UV para un objeto con forma cilíndrica y la asignación UV predeterminada.

Puede administrar el mapeo de texturas en una malla seleccionada con el Editor UV.
Veamos sus principales áreas de trabajo:

© 2023 Unity Technologies

Machine Translated by Google

1. Esta es la misma barra de herramientas que la de la vista de Escena con opciones para mover, rotar y escalar vértices, aristas o caras.

2. Estos botones le permiten manipular las coordenadas del mapeo UV directamente desde la vista de Escena cuando están habilitadas. Se colorean cuando están activas y se muestran en gris cuando no.
   a. El botón con flechas bloquea las herramientas de transformación en la vista de Escena para manipular el UV en lugar de la transformación del elemento seleccionado en la vista de escena (borde, vértice, cara u objeto).

   b. El botón “ladrillo” permite previsualizar la textura proveniente del sombreador en las coordenadas 0,0.

   c. El botón de la cámara mostrará la textura con el alambre poligonal superpuesto como referencia al editar la imagen en el software DCC. Por defecto, el archivo se guardará en la carpeta "Recursos" del proyecto.

3. Seleccione UV para editar el mapeo UV de los sombreadores. Seleccione UV2 (solo lectura) para regenerar sus mapas de luz predefinidos o en tiempo real.

4. Hay dos formas de trabajar en el mapeo UV del objeto seleccionado:

   a. Automático: ProBuilder gestiona el mapeo de texturas según la configuración del panel Acciones, incluso al redimensionar la malla. Esta es la opción predeterminada y probablemente suficiente para la mayoría de los diseños de niveles, especialmente si solo se trabaja con patrones repetitivos para prototipado.

   b. Manual: Este método permite desenrollar y editar elementos UV con precisión, renderizarlos y más. Se recomienda para posicionar los elementos UV en una imagen detallada. Puedes ver este tutorial . para una mirada paso a paso sobre texturizado avanzado con UV manuales.

5. Puedes manipular los elementos para organizarlos cuidadosamente de manera que coincidan. La textura del sombreador. En la imagen superior, una cara seleccionada se resalta en azul.

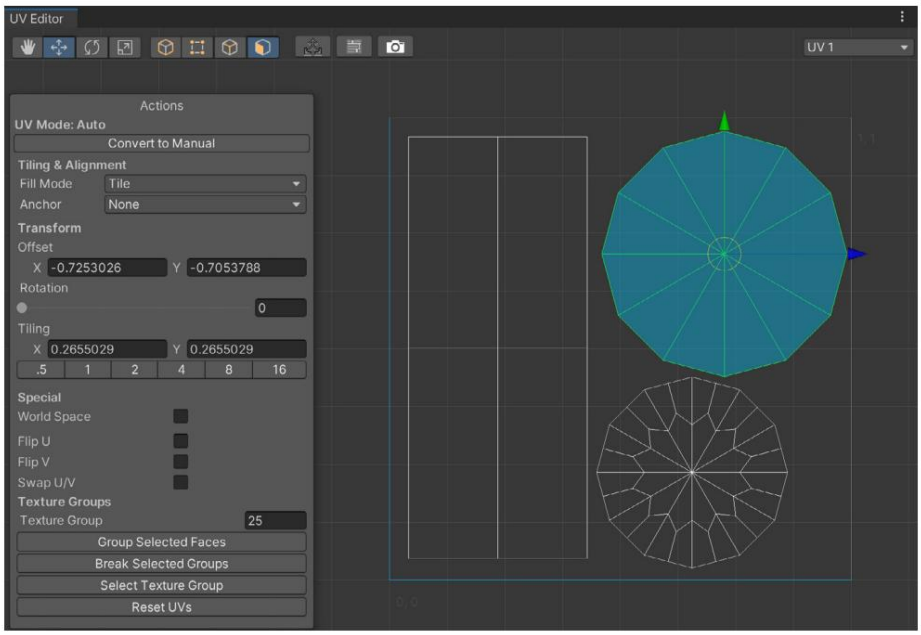
Ejercicio rápido de texturización

Al trabajar en modo automático , la agrupación de elementos, similar a un grupo de suavizado, permite manipular las partes agrupadas como un solo elemento. Esto resulta útil para mover, escalar o rotar el elemento seleccionado.

En este ejercicio, se crearon tres grupos para las caras superior, inferior y laterales del cilindro. Se utilizó una textura para este objeto y los UV se asignaron a ella. Para ello, se seleccionaron los tres grupos y se redistribuyeron dentro del área cuadrada que representa la textura.

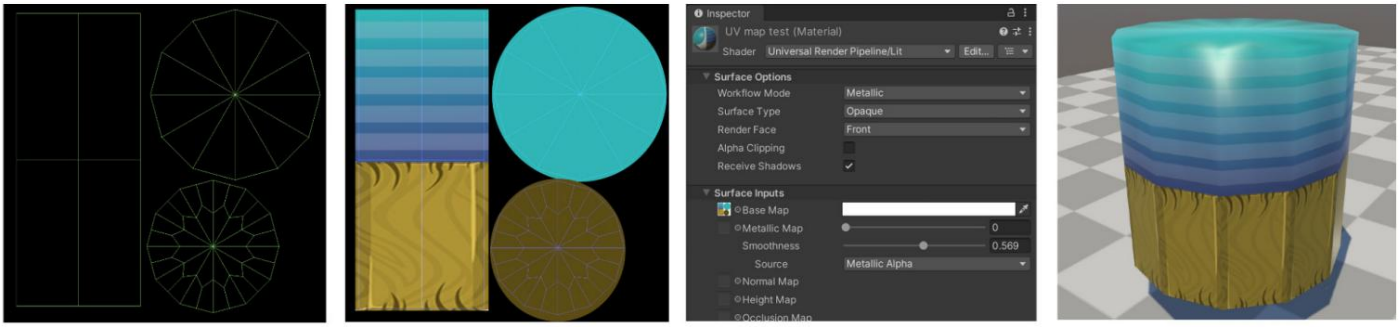
© 2023 Unity Technologies

Machine Translated by Google



Las caras en la parte superior del cilindro están agrupadas para que se puedan mover y escalar como una sola. La vista previa de la textura está deshabilitada, ya que esta se creará más adelante, basándose en el nuevo mapeo UV. En este ejercicio, si un diseñador no está satisfecho con los cambios, puede cambiarlo al mapeo UV predeterminado con la opción Restablecer UV .

Cuando el nuevo mapeo UV sea como debería ser, puedes exportar la textura para pintar sobre el objeto en tu herramienta DCC preferida.



Este es un ejercicio simple que muestra cómo texturizar rápidamente un objeto al crear un prototipo de diseño de niveles.

Repasemos cada uno de los pasos ilustrados en las imágenes anteriores, de izquierda a derecha:

Extremo izquierdo: el mapa de textura UV se importa para pintar en una herramienta DCC.
Centro izquierda: El modelo en alambre se utiliza como referencia al pintar.
Centro derecha: El archivo de imagen se devuelve a Unity y se crea un material con esta textura como Mapa Base. Puedes seguir iterando sobre una textura, siempre que esté dentro de tu proyecto de Unity y la use el material. Se actualizará automáticamente cada vez que guardes cambios en la imagen.

Extremo derecho: el material se aplica al objeto ProBuilder.

© 2023 Unity Technologies

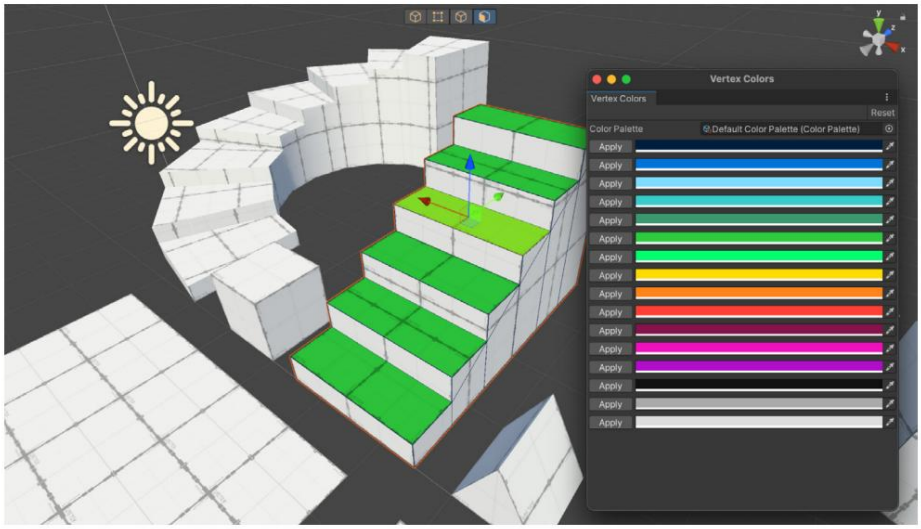
Machine Translated by Google

Consulte la documentación para obtener más información sobre cada opción disponible en el editor UV para Auto y Manual modos.

Consejo: Codificación de colores para un diseño de niveles rápido

La codificación por colores al delimitar los niveles puede ayudar a transmitir mejor su intención e ideas. claramente. Por ejemplo, puedes comunicarle al resto de tu equipo qué Los elementos son destructibles coloreándolos de rojo.

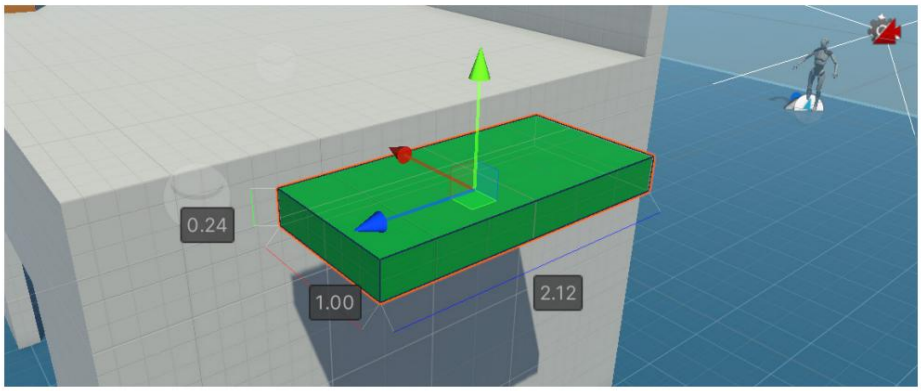
En la función Colores de vértice, seleccione el ícono “más” para crear una paleta de colores. Personaliza una paleta para definir los colores (y la cantidad de colores) que deseas. Para colorear un objeto, selecciónelo en la escena y haga clic en Aplicar. También puedes aplicar color a caras individuales y luego compartir tu paleta guardada de Colores con tu equipo para que todos usen el mismo código de colores normas.



Una escena en un cuadro gris codificado por colores

Consejo: Habilite las superposiciones de dimensiones para ProBuilder

En Herramientas > Probuilder > Superposición de dimensiones, puede habilitar etiquetas flotantes indicando el tamaño del objeto seleccionado actualmente en la escena.



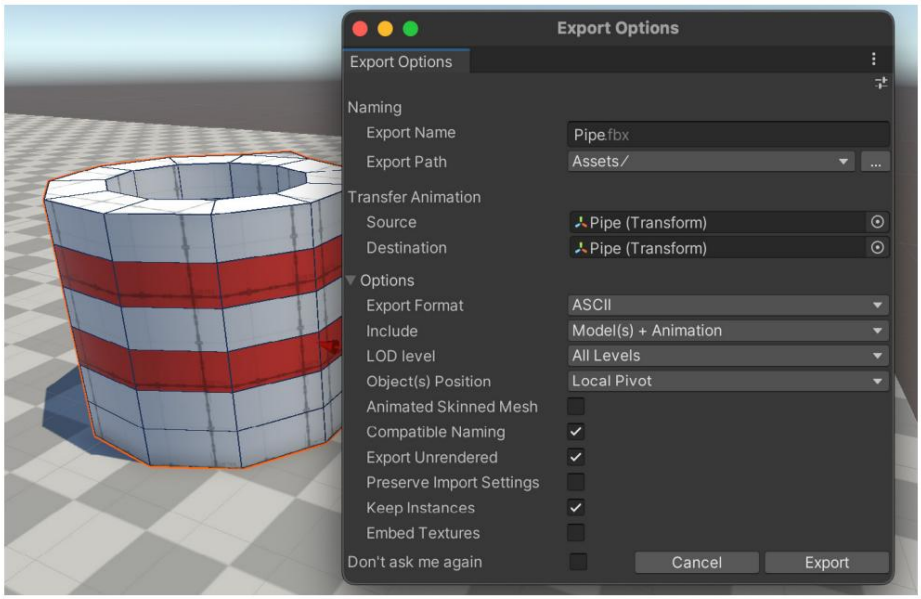
Visualización práctica de la altura, el ancho y la profundidad del objeto seleccionado

© 2023 Unity Technologies

Machine Translated by Google

Compartir niveles de ProBuilder con artistas ambientales

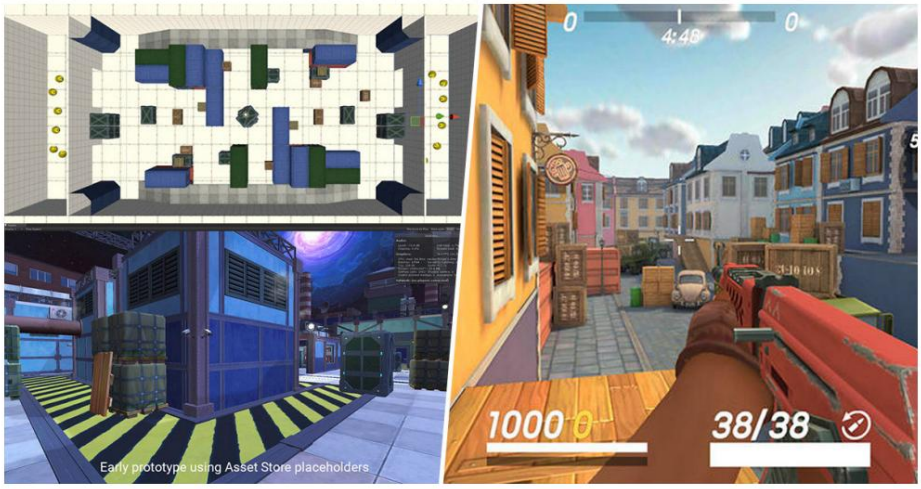
Si instala el Exportador FBX Paquete, podrás exportar tus recursos de nivel prototipados, con las dimensiones correctas, a una aplicación DCC para que un artista los pula y refine.



Cuando instales el paquete FBX Exporter, en GameObject > Exportar a FBX, tendrás opciones para exportar tu modelo y compartirlo con tus artistas.

Definir un plan de trabajo claro con el equipo permitirá un proceso de diseño fluido y eficiente en el que los artistas podrán trabajar en función del tamaño y la forma adecuados de los objetos del entorno 3D sin problemas.

También puedes usar las herramientas ProBuilder o Unity Asset Store para "kitbash", Esto implica combinar diferentes recursos para crear algo original y novedoso. La idea surge de los aficionados al modelismo, que combinan maquetas de trenes o aviones para construir sus propios proyectos personalizados.



Armas de auge (derecha) junto a prototipos (izquierda) creados a partir de activos de Unity Asset Store

© 2023 Unity Technologies

Machine Translated by Google

Pincel polivinílico es una herramienta de escultura 3D, similar a las herramientas de escultura de terreno, pero para mallas. Viene con modos Para combinar texturas, esculpir mallas con color y dispersar objetos directamente en el Editor de Unity. En combinación con ProBuilder, Polybrush te ofrece una solución completa de diseño de niveles en el Editor para que pruebes diferentes aspectos de tu entorno durante el proceso de diseño.

La ventana Polybrush se encuentra en la barra de herramientas superior, junto al menú de ProBuilder. Los principales modos de trabajo son:

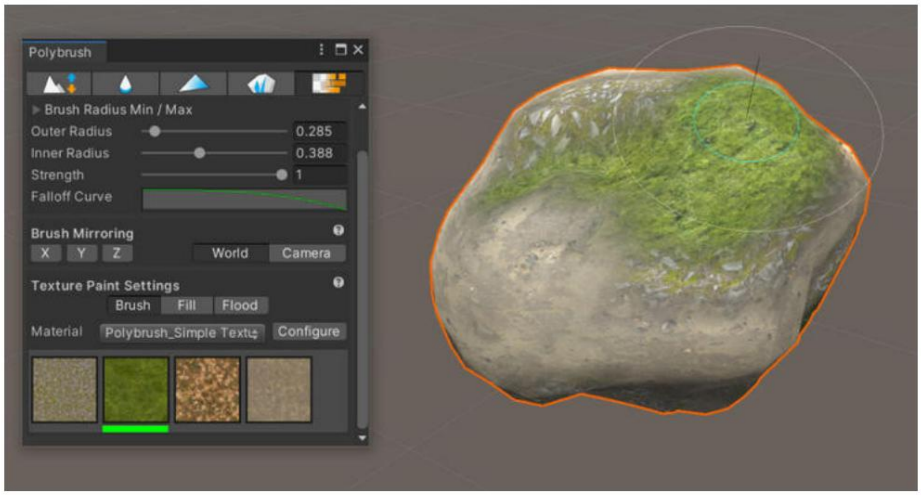
— Esculpir: utilice esta opción para empujar y tirar de los vértices y dar forma a la malla.

— Suave: promedia las diferencias entre las posiciones de los vértices a lo largo de un eje, que se establece mediante la propiedad Dirección en la configuración; use este modo para hacer que los vértices irregulares en la malla sean más uniformes

— Color: Establece los colores de los vértices en una malla con un pincel o un bote de pintura.

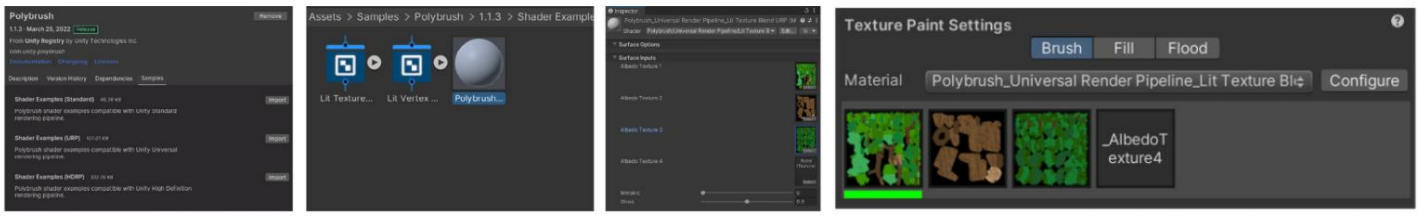
— Textura: Pinta y combina múltiples texturas en mallas

— Dispersión: coloca o dispersa prefabricados en la superficie de tu malla



Encuentra esta ventana en Herramientas > Polipincel > Polipincel para pintar texturas en mallas, entre otras opciones.

El modo de pintura de textura requiere los pasos adicionales que se muestran aquí, como asegurarse de que está utilizando un material compatible con la combinación de texturas en su malla.



Asegúrese de que su material utilice un sombreador que defina cómo combina las texturas.

© 2023 Unity Technologies

Machine Translated by Google

Repasemos los pasos que se muestran en las imágenes de arriba:

1. Vaya a la pestaña Muestras en la ventana del Administrador de paquetes de Polybrush. Obtenga el sombreador que corresponde a su canal de renderizado.
2. Haga clic derecho en el sombreador y elija Crear material.
3. Agrega las texturas que utilizará este material en su Inspector.
4. Polybrush detectará el material del objeto y le permitirá seleccionar Qué textura pintar. Ten en cuenta que quizás tengas que guardar los cambios primero.

Mira los siguientes videos para aprender a usar ProBuilder y Polybrush para diseñar, prototipar y terminar artísticamente una escena.

Crea un planeta con Polybrush

Iteración más rápida con ProBuilder y Polybrush

© 2023 Unity Technologies

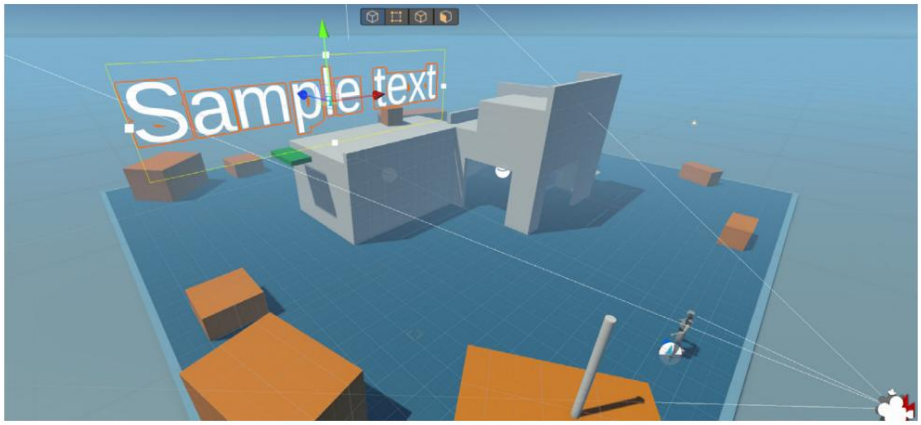
Machine Translated by Google

Consejo rápido: Visualizadores

Como se explicó anteriormente, puedes comunicar intenciones con tu equipo a través del color.
Codificar o usar bibliotecas de materiales en ProBuilder. Desarrollar juegos es complejo,
Y si puedes presentar tus ideas de prototipos lo más claramente posible a
Colegas, pueden tomar decisiones informadas sobre lo que funcionará y lo que no.
Aquí hay un par de consejos adicionales para comunicar claramente su diseño.
intención.

Texto de objeto 3D

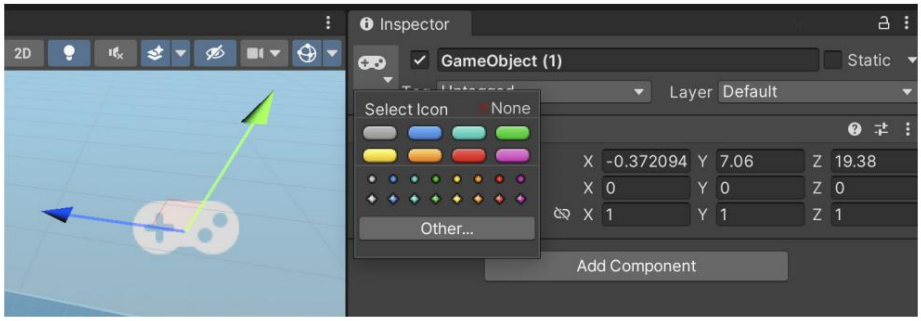
Cree letreros flotantes en el espacio 3D a través de GameObject > Objeto 3D > Texto >
TextMeshPro. Estos pueden actuar como notas adhesivas virtuales que proporcionan información adicional.
Información a tus colegas (y a ti mismo) sobre el diseño de niveles. Muévete,
escalarlos y agruparlos bajo un GameObject padre para fines organizativos.



El objeto de texto 3D es fácil de mover y cambiar de tamaño en un diseño de nivel de caja blanca o gris porque es independiente.
elemento que no es parte de un sistema de interfaz de usuario como las etiquetas de texto normales.

Icono personalizado para GameObjects

Los íconos personalizados hacen que los GameObjects sean fáciles de identificar, especialmente en un juego grande
mundo. Ejemplos de cómo usar íconos personalizados se encuentran al principio y al final de un nivel, por ejemplo.
Puntos de aparición de jugadores y enemigos, y áreas ocultas o secretas. Los íconos personalizados pueden
También se puede utilizar para transmitir información rápidamente, como la salud o el estado de un nivel.
Jefe. Puedes cambiar el ícono desde el Inspector, junto al nombre del GameObject.
Obtenga más información sobre las posibilidades de visualización de gizmo en este tutorial de Unity Learn.



Personalizar la apariencia del icono de GameObject que se muestra en la vista de escena

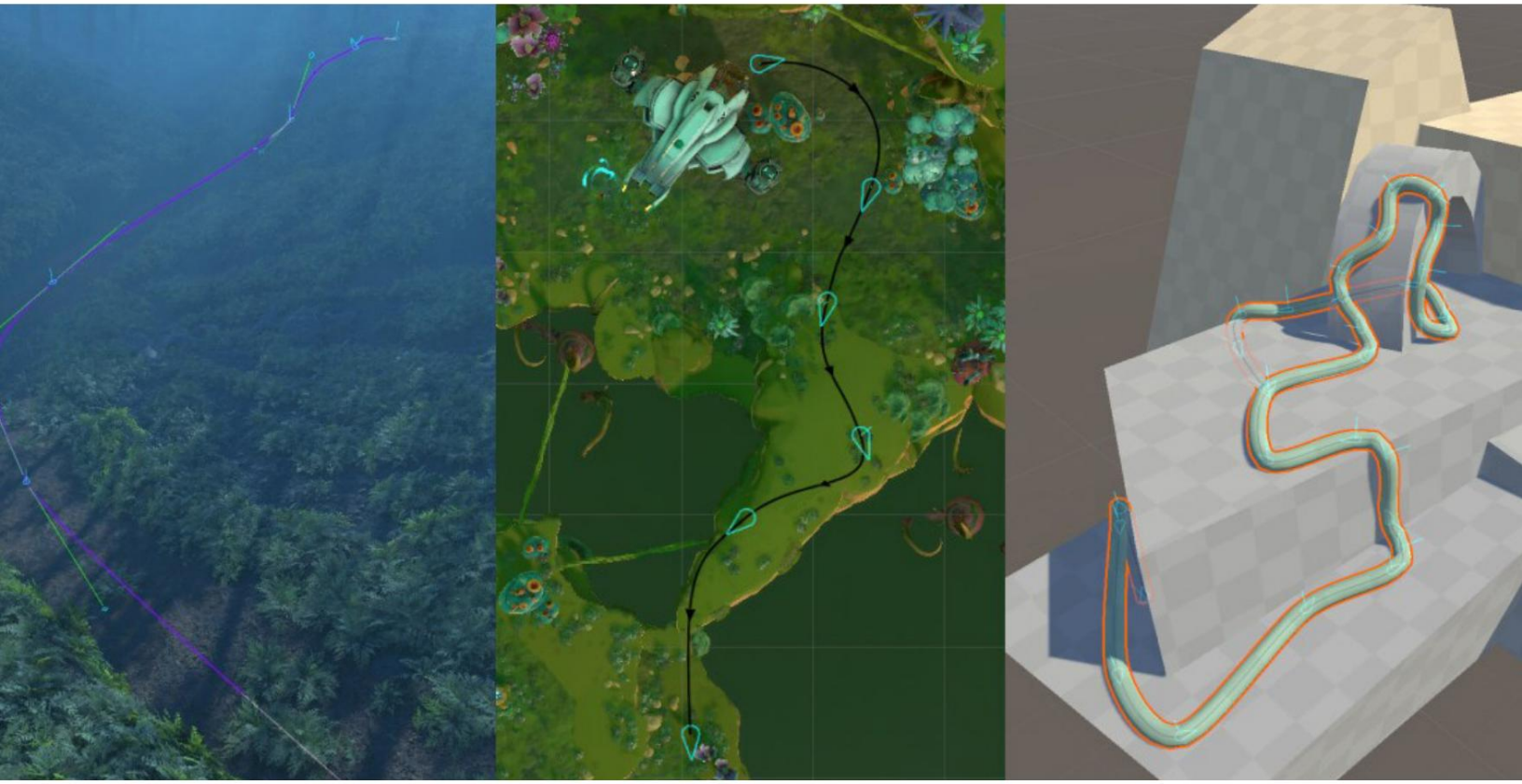
© 2023 Unity Technologies

Machine Translated by Google

Diseño e implementación de interfaz de usuario en Unity

Desarrolla tus habilidades aprendiendo a usar las herramientas de creación de UI en Unity. En esta guía de más de 130 páginas para artistas, diseñadores y desarrolladores, encontrarás consejos y prácticas recomendadas para crear interfaces sofisticadas con los dos sistemas de UI de Unity: Unity UI y UI Toolkit.

Descargar el libro electrónico



Ejemplos de casos de uso para splines: un camino a través de un bosque, una ruta de animación, tubos o mallas de alambre; encuentre otros ejemplos que muestran todas las posibilidades en la página del Administrador de paquetes de splines una vez que esté instalado

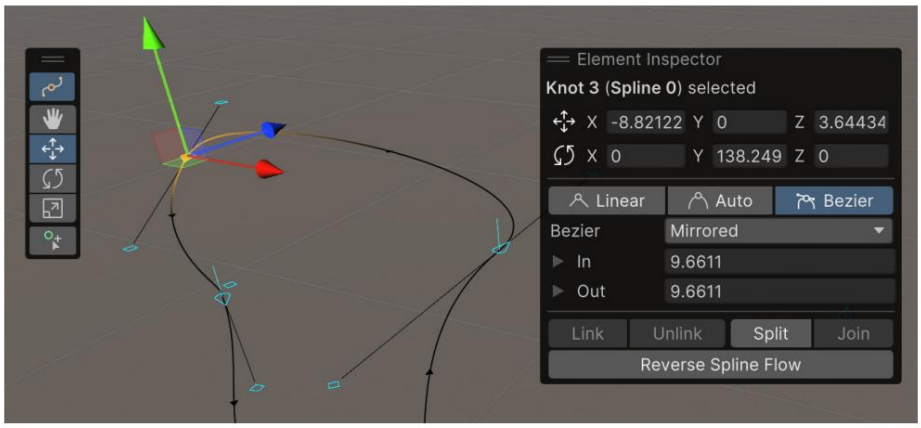
Estrías

Introducido en Unity 2022 LTS, el paquete Splines Te permite crear rutas spline en tu juego para ríos, carreteras, pistas de cámara y otros elementos relacionados con las rutas. Dependiendo del tipo de entorno en el que trabajes, las splines pueden ser un componente importante para el diseño de tus niveles.

© 2023 Unity Technologies

Machine Translated by Google

Cree un nuevo spline mediante GameObject > Spline > Herramienta Dibujar Splines. Se creará un nuevo GameObject en la jerarquía con el componente Spline adjunto y las herramientas listas para usar.



Los controladores y controles para splines en Unity se parecen a las herramientas de dibujo vectorial o 3D de las conocidas aplicaciones DCC.

Una vez creado un spline, tanto programadores como artistas pueden usarlo de maneras interesantes. Por ejemplo, los programadores pueden leer sus puntos y usarlos en la lógica del juego con las API.

Los componentes disponibles para aprovechar splines son:

— Instanciación de spline: Genera copias de un elemento a lo largo de una spline. Usa el componente Instanciación para crear objetos como vallas, árboles, caminos de piedra, etc.

— Animación por spline: Mueve un GameObject a lo largo de una spline. Usa el componente Animate con cámaras, personajes o en situaciones donde necesites definir movimiento en Unity.

— Extrusión de spline: Construye una malla tubular a lo largo de una spline. Usa el componente Extrusión para crear y editar formas como cables, tuberías, cuerdas, fideos y más.

Obtenga más información sobre las próximas funciones nuevas para Splines en esta publicación de blog.

© 2023 Unity Technologies

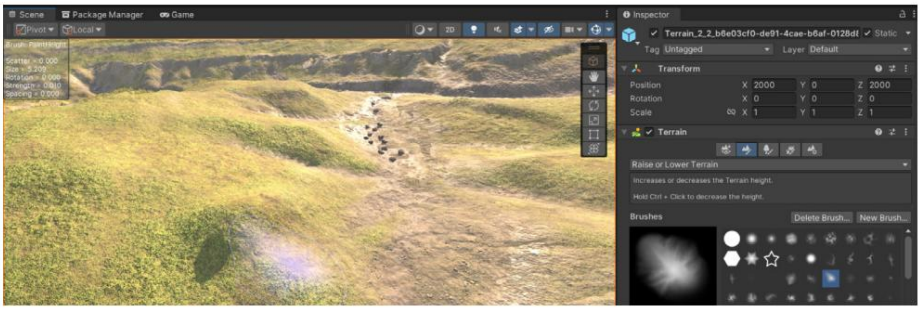
Machine Translated by Google

Los pinceles y sellos de terreno te ayudan a crear paisajes realistas. Como se mencionó anteriormente, los datos SIG pueden ser una excelente fuente de inspiración para el diseño de terrenos.

Terreno

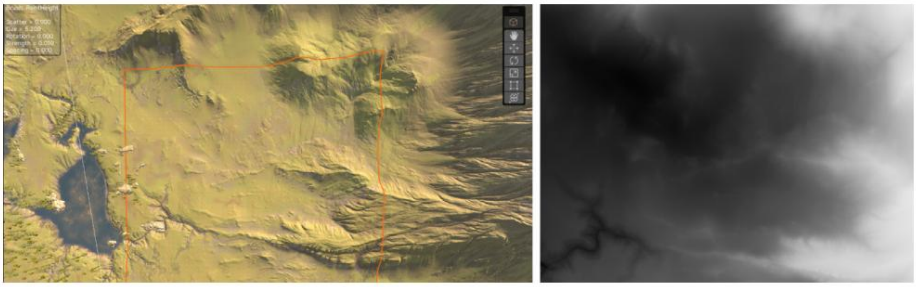
El editor de terreno de Unity Permite crear terrenos realistas y optimizados. Es una herramienta eficaz para crear superficies orgánicas y no planas, y puede usarse en combinación con las demás herramientas de diseño de niveles descritas en secciones anteriores.

Para empezar a usar las herramientas de Terreno, crea un objeto de terreno en GameObject > Objeto 3D > Terreno. El componente Terreno proporciona pinceles para elevar o bajar el terreno donde pintes el mapa de alturas con la herramienta Pincel. Úsalo para ocultar partes del terreno, añadir un pincel de sello sobre el mapa de alturas actual o simplemente pulirlo.



Usar un pincel para crear algunas elevaciones en el terreno; tenga en cuenta que el Editor de terreno tiene como objetivo modificar terrenos a gran escala, lo que significa que las configuraciones predeterminadas generalmente utilizan números grandes de unidades de Unity.

Los cambios en la superficie se reflejan en la geometría de la superficie del terreno con colisionadores, texturas o vegetación o árboles adjuntos que reaccionan a los cambios automáticamente.



Puedes exportar un mapa de alturas generado en el Editor de Terreno, como se muestra en la imagen de la derecha, que corresponde al terreno con borde rojo en la imagen de la izquierda, o importar mapas de alturas externos. En este video de la comunidad , puedes aprender a importar datos SIG o del mundo real a Unity.

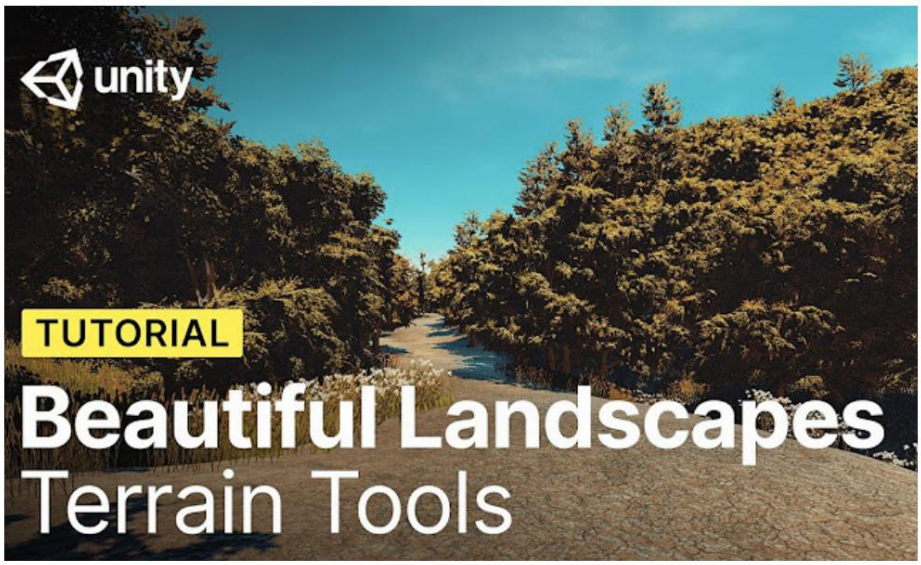
© 2023 Unity Technologies

Machine Translated by Google

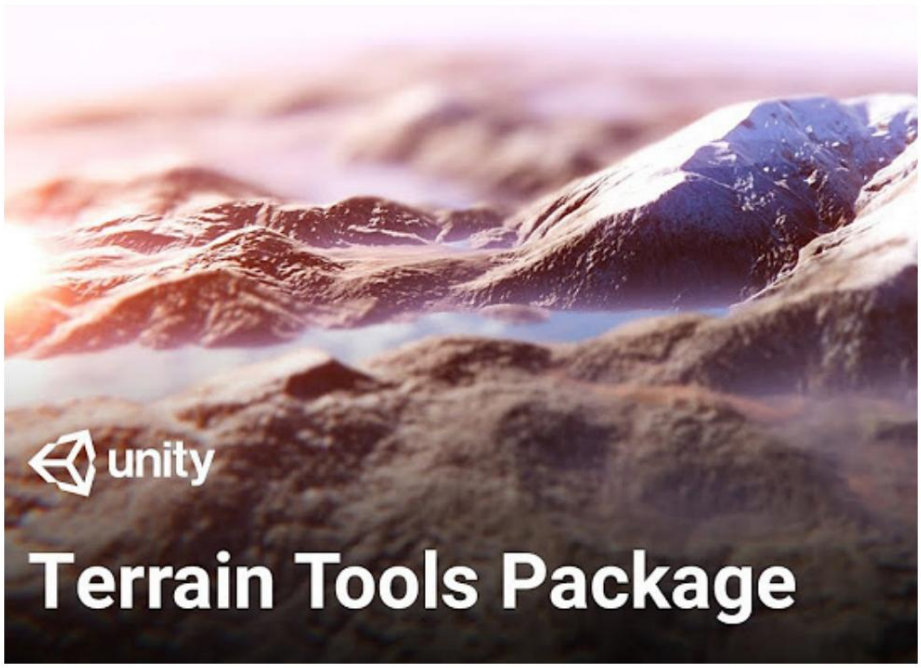
El Editor de Terreno incluye configuraciones para definir distancias de dibujo de vegetación, árboles o terreno, además de configuraciones para controlar la resolución de la malla del terreno, la iluminación y los efectos del viento.

Junto con el Editor de Terreno, puedes instalar las Herramientas de Terreno paquete que agrega pinceles y herramientas de escultura de terreno adicionales a su proyecto para crear activos de terreno vívidos.

Aprende más sobre las herramientas de terreno en Unity con los siguientes videos. También puedes importar la escena de ejemplo en el Gestor de Paquetes para aprender cómo se creó una escena final.

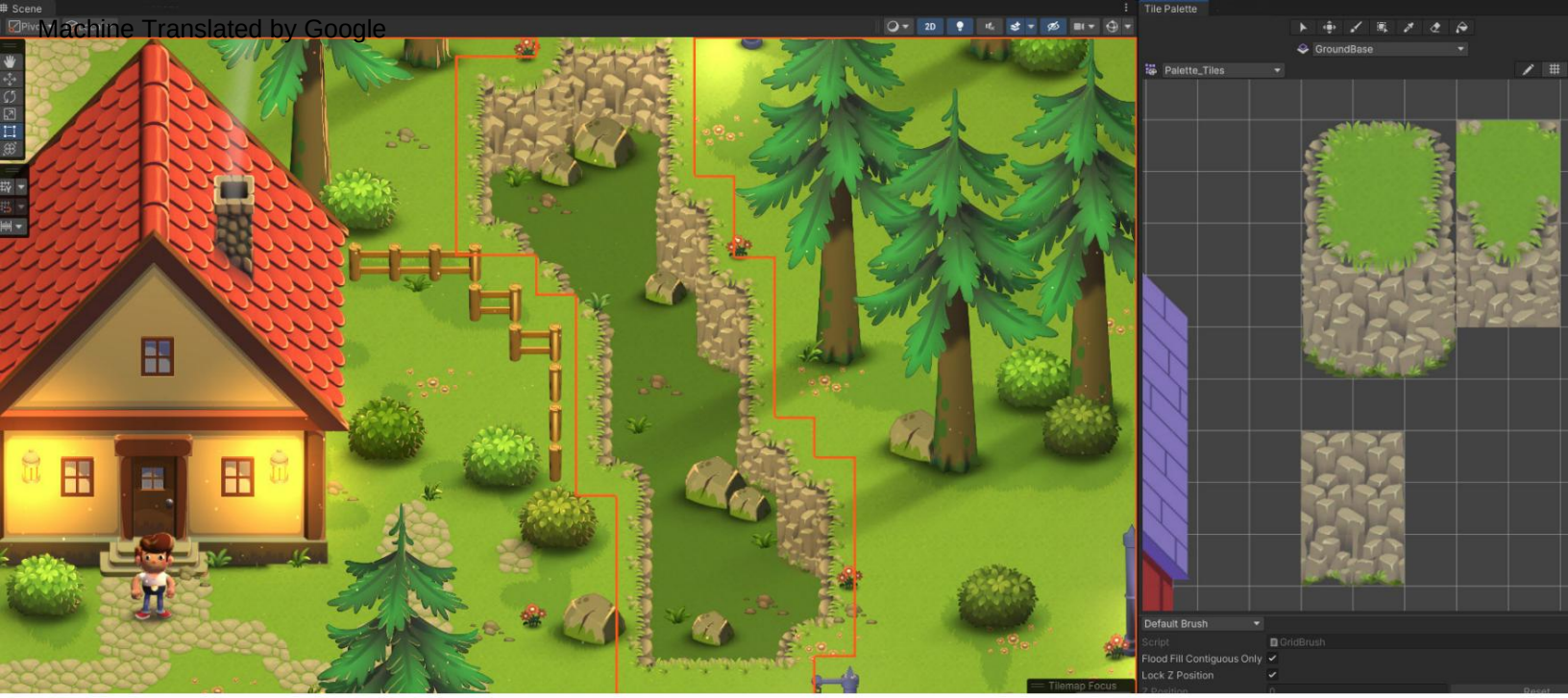


Hermosos paisajes con Terrain Tools



Vídeo del paquete de herramientas de terreno

Machine Translated by Google



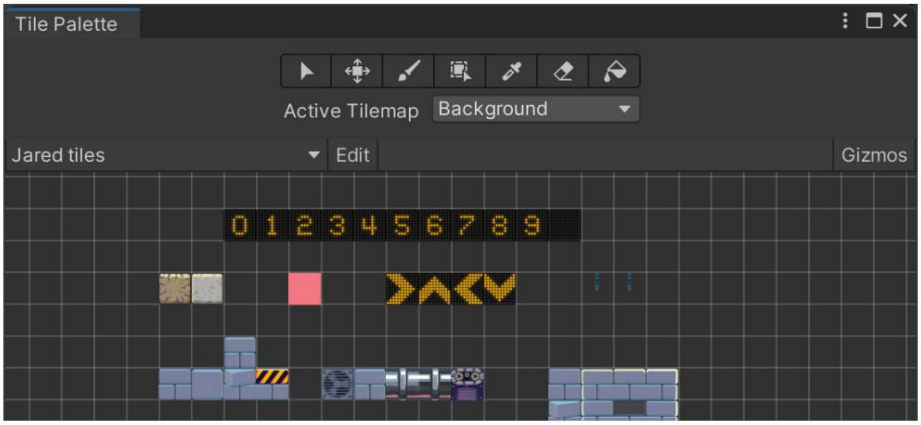
Uso de mapas de mosaicos para crear un juego de arriba hacia abajo

Mapa de mosaicos 2D

El editor de mapas de mosaicos 2D se instala con la plantilla de proyecto 2D o desde el Administrador de paquetes. Mapa de mosaicos 2D Es una excelente función para prototipado rápido. Tilemap permite crear un mundo de juego usando pequeños sprites, llamados tiles, colocados en una cuadrícula. En lugar de diseñar un mundo de juego como una sola imagen grande, puedes dividirlo en bloques que se repiten a lo largo de todo el nivel. Tilemap 2D admite tiles rectangulares, isométricos y hexagonales.

La herramienta de pincel facilita la pintura de mosaicos en una cuadrícula, y se pueden agregar reglas de pintura mediante scripts. También incluye generación automática de colisiones para una prueba y edición eficientes.

La ventana de la paleta de mosaicos contendrá todos los mosaicos y herramientas para ayudarte a pintar o editar mapas de mosaicos. Crea una nueva paleta haciendo clic en el botón "Crear nueva paleta". Aparecerá un menú desplegable con opciones. Dale un nombre a la paleta, configura sus opciones, haz clic en "Crear" y guárdala en la carpeta seleccionada.

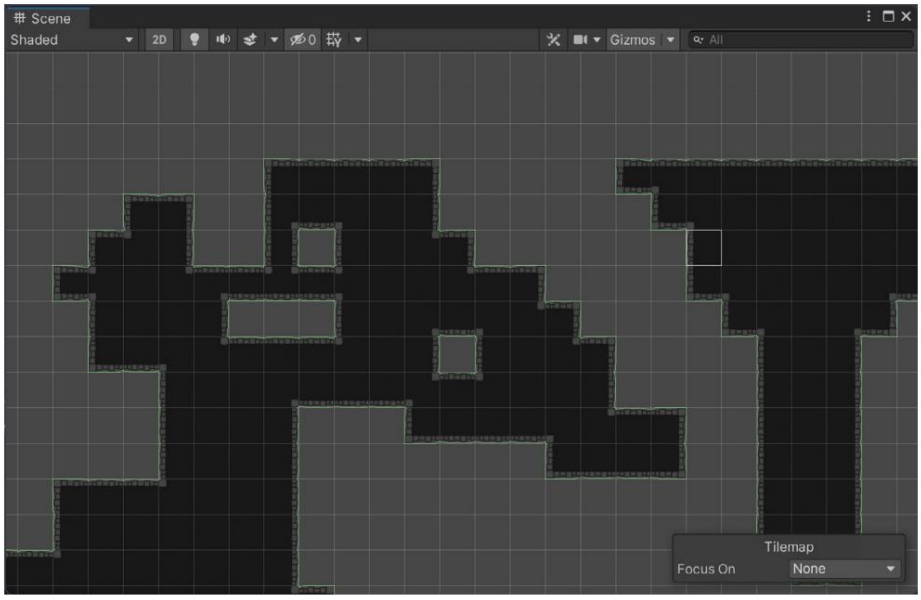


Paleta de mosaicos ventana con una paleta cargada

© 2023 Unity Technologies

Machine Translated by Google

Ahora, agrega mosaicos a la paleta. Arrastra un objeto (se te pedirá que crees un mosaico).
Activo) o Activo de mosaico en la ventana Paleta. Ahora deberías ver tu sprite en
la cuadrícula de la paleta.



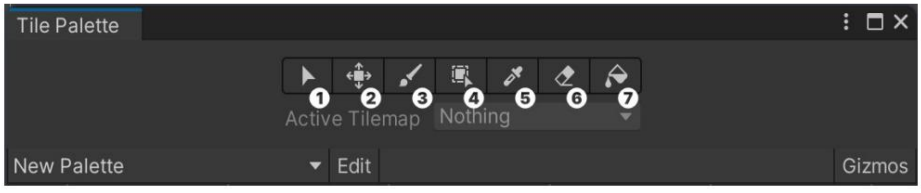
Un prototipo de un nivel basado en cuadrícula con mosaicos de marcador de posición; el contorno verde muestra los límites del colisionador

Los mapas de mosaicos te permiten prototipar un nivel 2D de forma eficiente. Añade un colisionador de mapas de mosaicos.
Componente 2D al mapa de mosaicos para física. Esto añadirá colisionadores a cada uno de los...
mosaicos basados en el tipo de colisionador establecido en el recurso de mosaico.

Puede agregar un colisionador compuesto 2D para combinar los colisionadores en uno, lo que da como resultado
En un comportamiento de colisión más suave con la geometría establecida como Contorno. Los mosaicos...
se comportan como un terreno continuo en lugar de como mosaicos individuales. Esta configuración también
ofrece una ligera mejora del rendimiento. Sin embargo, si planeas que tu juego agregue o
eliminar mosaicos en tiempo de ejecución, es posible que desees mantener un colisionador en cada mosaico.

Recuerde marcar la opción Usado por Compuesto en el Colisionador 2D
componente y configure el tipo Rigidbody 2D en estático para que no se caiga.

Veamos las diferentes herramientas de la Paleta (los atajos de teclado se indican en
corchetes).



Herramientas de paleta de mosaicos

© 2023 Unity Technologies

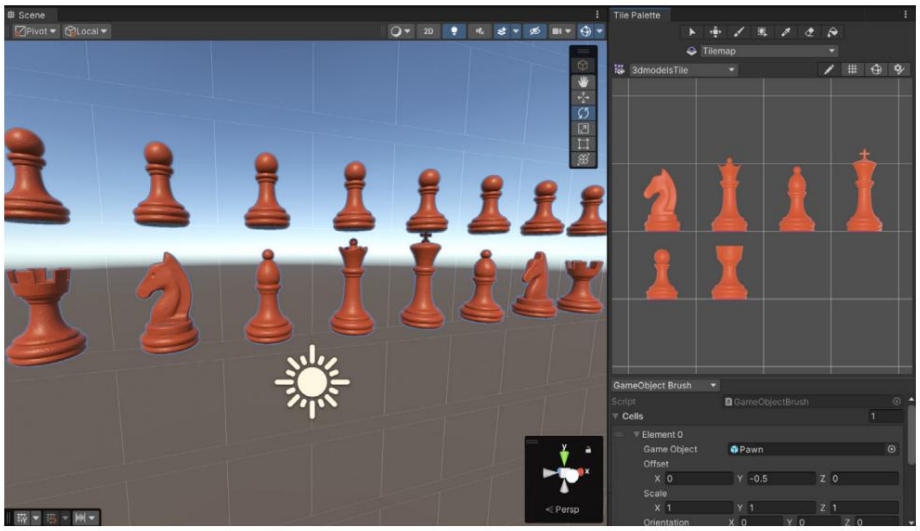
Machine Translated by Google

1. Selección (S): Haga clic para seleccionar un mosaico o arrastre para seleccionar mosaicos dentro de un área rectangular
2. Mover (M): Mueve los mosaicos seleccionados
3. Pincel (B): pinta sobre un mapa de mosaicos activo (selecciona uno del menú desplegable Mapa de mosaicos activo) con el mosaico y el pincel seleccionados
4. Rellenar selección (U): arrastre para rellenar un área rectangular usando un mosaico seleccionado
5. Muestreador de mosaicos (I): seleccione un mosaico de un mapa de mosaicos y configúrelo como activo para pintar
6. Borrador (D): elimina mosaicos de un mapa de mosaicos
7. Rellenar (G): rellena un área con un mosaico (el área debe estar bordeada con otros mosaicos)

Extras del mapa de mosaicos 2D

Un paquete adicional, 2D Tilemap Extras, Contiene scripts adicionales para Tilemaps que habilitan nuevas funciones como mosaicos de reglas, mosaicos animados y más.

Una característica destacada para los diseñadores de niveles es la compatibilidad con GameObjects, que permite crear una paleta de Tiles compuesta por cualquier Prefab, como un modelo 3D, y usar la cuadrícula del mapa de Tiles 2D para organizar dichos GameObjects en la escena. Esta puede ser una potente herramienta de diseño de niveles cuando el juego necesita seguir un patrón de cuadrícula.

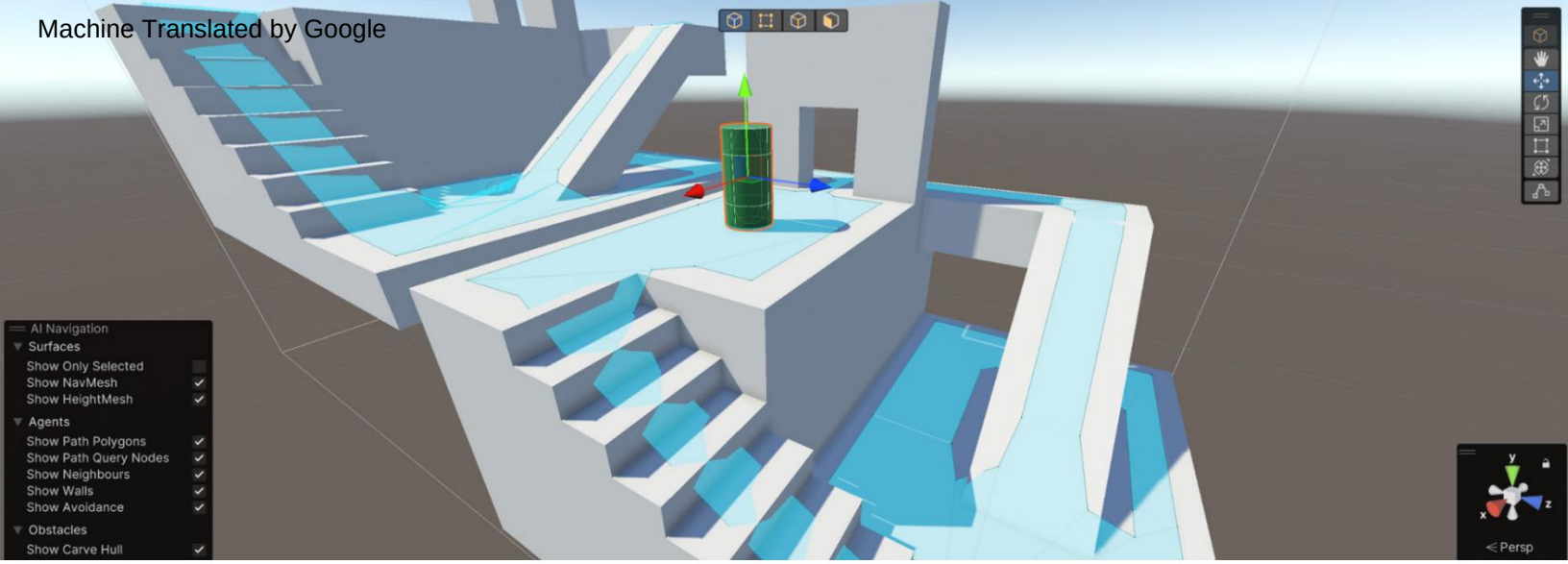


Uso de Tilemap 2D para organizar GameObjects en la escena

Para crear tu paleta de mosaicos con GameObjects, cambia la opción Pincel predeterminado del menú desplegable a Pincel de GameObject y arrastra un prefab al campo Celdas > Elemento 0 > GameObject. Usa el pincel para añadir este prefab a la paleta de mosaicos. Al terminar, cambia el campo GameObject por otro prefab y añádelo a la paleta de mosaicos. Repite este proceso para todos los GameObjects que quieras añadir a la paleta de mosaicos.

© 2023 Unity Technologies

Machine Translated by Google

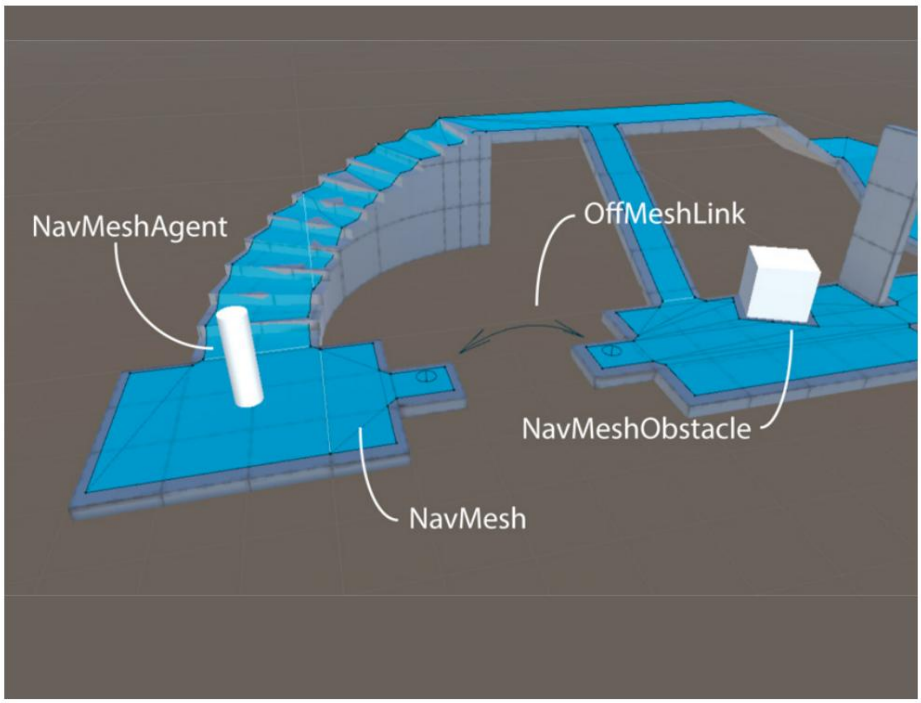


Búsqueda de rutas sencilla para personajes o NPC con navegación IA

Búsqueda de rutas con navegación de IA

Es muy probable que tu jugador u otros NPC necesiten navegar por el mundo para llegar un destino. Además de la mecánica del juego, ver a los personajes navegar por sí mismos El mundo puede ayudar a sumergir a los jugadores en tu juego.

La navegación de IA El paquete (anteriormente conocido como NavMesh) le permite crear personajes (llamados agentes en el contexto de la navegación de IA) que pueden moverse por el mundo del juego, utilizando mallas de navegación que se crean automáticamente desde la geometría de su escena.



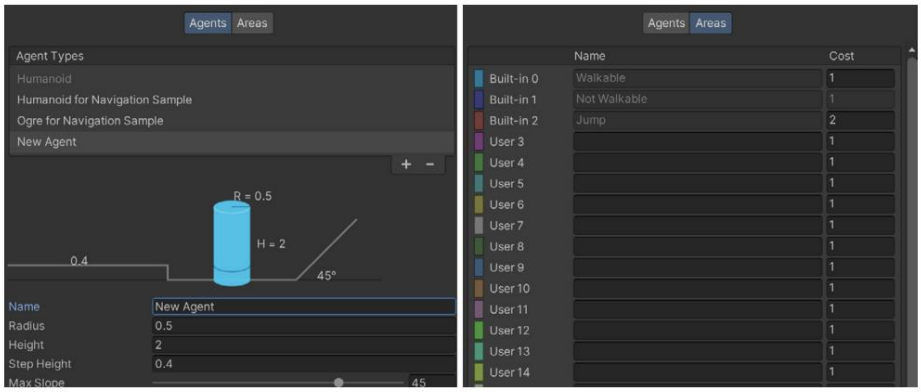
Componentes clave de AI Navigation: un agente NavMesh, una superficie NavMesh, un obstáculo NavMesh y un enlace NavMesh

© 2023 Unity Technologies

Machine Translated by Google

La ventana de Navegación: Agentes y áreas

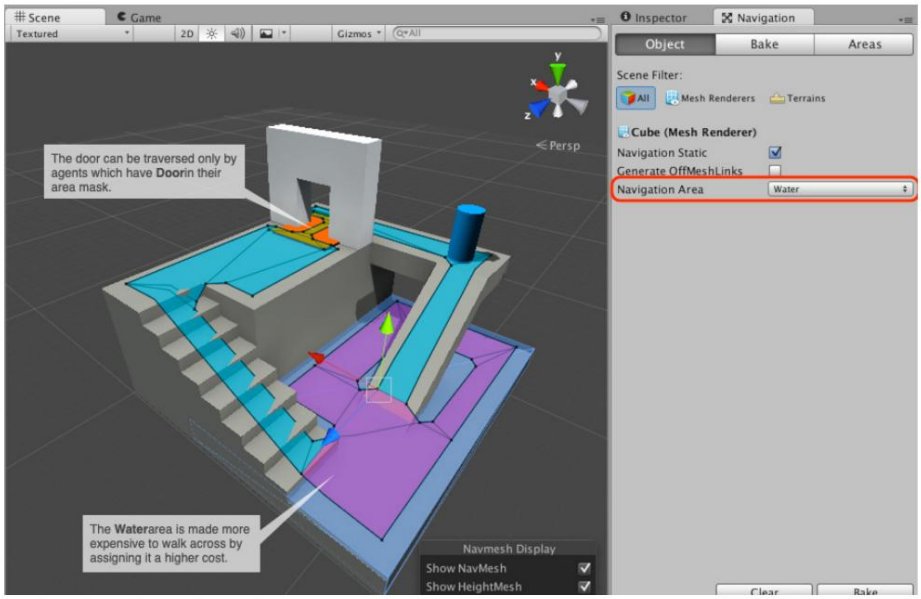
En el menú superior, en Ventana > IA > Navegación, puedes configurar los agentes o los tipos de personajes que navegarán por el mapa y las áreas que son las rutas preferidas para los agentes.



Definición de agentes y áreas para un proyecto

El tipo de agente Define el volumen general del personaje y, al igual que el Colisionador de Personajes, se representa con una forma cilíndrica. La altura del paso define la distancia vertical máxima que el agente puede recorrer, y la pendiente máxima, como su nombre indica, establece el límite de la pendiente que el agente puede ascender. Los agentes también pueden saltar de una superficie de malla a otra, lo que permite al personaje caer a un nivel inferior o saltar a otra zona, por ejemplo. La altura de caída y la distancia de salto definen estos parámetros.

Los algoritmos de navegación de IA utilizan las configuraciones del tipo de agente cuando prepara el área navegable en el componente Superficie NavMesh usando el botón Hornear .



Un ejemplo de la documentación de Unity que muestra un área de mayor costo en color púrpura

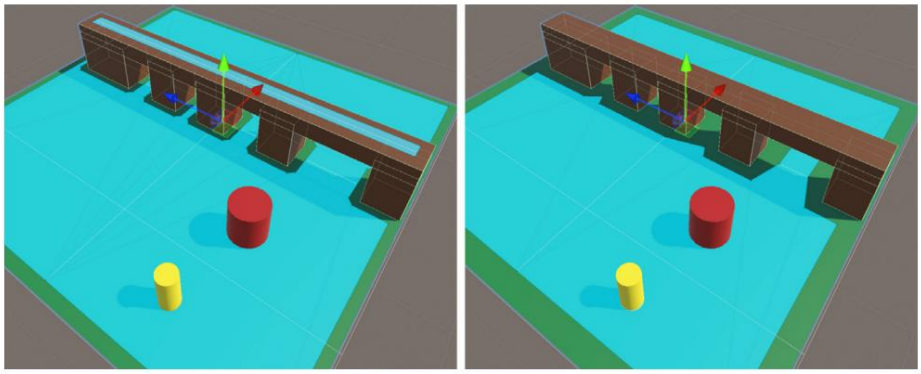
© 2023 Unity Technologies

Machine Translated by Google

Las áreas Define la dificultad de caminar por un área específica; las áreas de menor costo se priorizarán durante la búsqueda de rutas. Además, cada agente de NavMesh tiene una máscara de área que permite especificar las áreas en las que puede moverse.

Utilice el componente NavMeshModifier Volume Para crear un área cúbica que se puede modificar para definir el tipo de área. Con el NavMeshModifier componente (que permite anular el área) para crear un objeto de malla particular de un determinado tipo de área o NavMeshLink Crea un enlace navegable entre dos ubicaciones. Este enlace puede tener un coste correspondiente a un tipo de área. Los diferentes tipos de área mostrarán diferentes colores de previsualización que puede configurar desde este menú.

Tres pasos para crear rutas navegables con AI Navigation



En la imagen de la izquierda, se genera la superficie NavMesh para el agente de color amarillo. Observe cómo la ruta transitable (área azul) le permite pasar por debajo de cada pilar del puente. A la izquierda, se genera la superficie NavMesh para el agente de color rojo. Debido a su tamaño, su ruta transitable solo le permite pasar por los espacios más amplios entre los pilares.

1. Cree una superficie NavMesh: defina la superficie por la que pueden navegar los agentes.
   Una forma rápida de hacerlo es mediante GameObject > AI > NavMesh Surface.
   Seleccione "Bake" para generar los datos de navegación según los parámetros predeterminados. Algunos ajustes clave del componente NavMeshSurface son:

   a. Tipo de agente: Cree una NavMeshSurface por cada tipo de agente que planee navegar por la superficie. Cada agente usará datos preconfigurados diferentes según su configuración. Si realiza cambios en el tipo de agente, deberá volver a configurar los datos.

   b. Área predeterminada: el tipo de área predeterminado de la superficie navegable generada; como se mencionó anteriormente, hay modificadores que anularán el tipo predeterminado elegido aquí.

   c. Generar enlaces: cuando está habilitado, se generarán enlaces de navegación para permitir que el personaje salte o caiga según los parámetros del agente Distancia de salto y Altura de caída.

© 2023 Unity Technologies

Machine Translated by Google

d. Usar geometría: puedes usar la información de la malla o los datos del colisionador físico para generar la superficie.

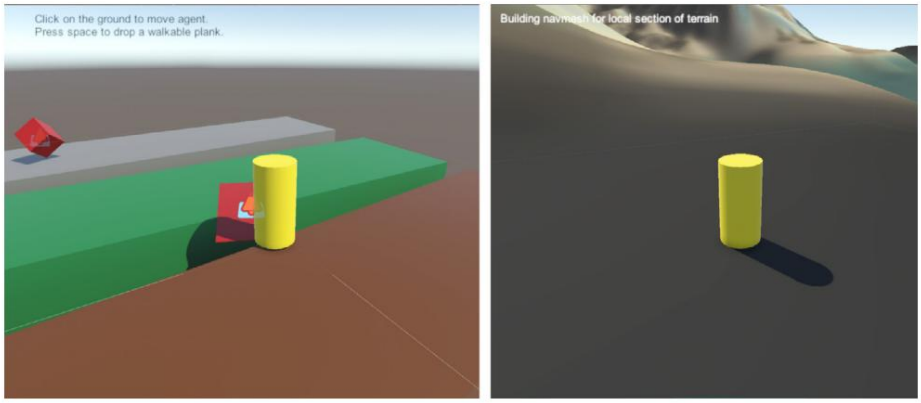
e. Colección de objetos: Permite filtrar los tipos de objetos o capas que se usarán para la generación de superficies. Por defecto, se usarán todas las mallas de la escena. Si desea excluir la malla de un objeto en particular, puede agregar los componentes "Obstáculo de Malla de Navegación" o "Modificador de Malla de Navegación" (eliminar objeto).

f. Avanzado: Esto le brinda la opción de ajustar la precisión requerida para los datos de navmesh.

g. Bake: Esto procesará los datos según las configuraciones aquí y la configuración del tipo de agente, mientras que Clear limpiará los datos.

2. Crea un GameObject que represente al personaje. Puedes seguir el ejemplo anterior y crear un cilindro que corresponda al tamaño del tipo de agente y asignarle el Agente NavMesh. componente. Asegúrese de que el agente esté configurado en el tipo que coincida con la superficie NavMesh.

3. Prueba la navegación. Puedes añadir el componente ClickToMove , incluido en los ejemplos de Navegación IA, al GameObject para que se mueva donde hagas clic. También puedes ver el comportamiento de NavigationGoals para que los PNJ se muevan de un punto a otro mediante Navegación IA.



Descargue e importe las muestras de navegación de IA desde el Administrador de paquetes

AI Navigation es compatible con ProBuilder y las herramientas de terreno, y también puedes visualizar las diferentes áreas navegables de la superficie y otra información relacionada con el tipo de agente con las herramientas de visualización.

© 2023 Unity Technologies

Machine Translated by Google

TileWorldCreator 3 disponible en la Asset Store

Referencia rápida de herramientas de diseño de niveles

La siguiente lista consta de las herramientas cubiertas en las secciones anteriores y Sugerencias de la tienda de recursos de Unity (organizadas por categoría). Ver esto sección de Unity Asset Store para soluciones especializadas para sus necesidades particulares género del juego.

Herramienta de Unity	Descripción	Dónde conseguirlo
Modelado 3D, creación de prototipos		
ProBuilder	Construir, editar y texturizar geometría directamente en Unity	Administrador de paquetes
Pincel polivinílico	Pintura, escultura y mallas dispersión geográfica	Administrador de paquetes
Estrías	Construir rutas spline y tubulares mallas 3D	Administrador de paquetes
UModeler	Modelado 3D de baja poli y prototipado de entornos	Tienda de activos de Unity

© 2023 Unity Technologies

Machine Translated by Google

SabreCGS	Construye niveles complejos usando Geometría sólida constructiva técnicas	Tienda de activos de Unity o GitHub
Generación de terreno basada en mapas de altura		
Terreno + Herramientas de terreno	Crea grandes paisajes con Azulejos de terreno; instala los extras para más pinceles y herramientas	Administrador de paquetes
Gaia Pro	Terreno y entorno herramientas de generación	Tienda de activos de Unity
Editor de terreno del Atlas	Procedimiento no destructivo creación de terreno	Tienda de activos de Unity
Diseño basado en cuadrícula y mosaicos		
Mapa de mosaicos 2D + Extras del mapa de mosaicos	Crea niveles 2D usando mosaicos activos dispuestos en una cuadrícula; También admite hexagonal y Azulejos isométricos e incluye tipos de mosaicos especiales, como los de regla mosaicos y mosaicos animados	Administrador de paquetes
IntelliMap Pro	Mapas de mosaicos generados por IA en 2D	Tienda de activos de Unity
Creador de TileWorld	Crear mapas de mosaicos 3D procedimental o manualmente	Tienda de activos de Unity
Tessera Pro	Generar niveles 3D basados en Prefabricados de azulejos	Tienda de activos de Unity
Organización, colocación y encaje de prefabricados		
Octave3D	Colección de prefabricados a presión y herramientas de pintura	Tienda de activos de Unity
Constructor inteligente	Organización de activos y productividad	Tienda de activos de Unity
Placer 2	Generación de objetos en el Editor o en tiempo de ejecución	Tienda de activos de Unity
Agarrar	Colocar objetos con física en Editor	Tienda de activos de Unity

© 2023 Unity Technologies

Machine Translated by Google

Recursos adicionales de diseño de niveles

No faltan canales de YouTube dedicados a temas de diseño de niveles.
Algunos buenos ejemplos que vale la pena revisar incluyen Game Maker's Toolkit, Steve Lee
(Diseño de niveles y juegos), y el vestíbulo de diseño de niveles.

Los foros de usuarios de Unity a los que los diseñadores de niveles pueden beneficiarse al unirse incluyen:
Prefabricados de creación de mundos Unity y Cinemachine Unity.

Finalmente, puedes encontrar todos los libros electrónicos avanzados de Unity para artistas, diseñadores y
programadores, además de muchos otros recursos valiosos en el centro de mejores prácticas de Unity.

Capacitación profesional para creadores de Unity

La capacitación profesional de Unity le brinda las habilidades y el conocimiento para trabajar más
de forma productiva y colaborando eficientemente en Unity. Encuentra una amplia formación.
Catálogo diseñado para profesionales de cualquier industria, en cualquier nivel de habilidad, en múltiples
formatos de entrega.

Todos los materiales son creados por diseñadores instruccionales experimentados en colaboración
con nuestros ingenieros y equipos de producto. Esto significa que siempre recibirá lo mejor.
La capacitación más actualizada sobre la última tecnología de Unity.

Más información sobre cómo Unity Professional Training puede ayudarle a usted y a su equipo.

© 2023 Unity Technologies

Unity®